Contents

1

A Survey of Collectives

Kagan Tumer and David Wolpert

ABSTRACT Due to the increasing sophistication and miniaturization of computational components, complex, distributed systems of interacting agents are becoming ubiquitous. Such systems, where each agent aims to optimize its own performance, but where there is a well-defined set of system-level performance criteria, are called **collectives**. The fundamental problem in analyzing/designing such systems is in determining how the combined actions of self-interested agents leads to "coordinated" behavior on a large scale. Examples of artificial systems which exhibit such behavior include packet routing across a data network, control of an array of communication satellites, coordination of multiple deployables, and dynamic job scheduling across a distributed computer grid. Examples of natural systems include ecosystems, economies, and the organelles within a living cell.
No current scientific discipline provides a thorough understanding of the relation between the structure of collectives and how well they meet their overall performance criteria. Although still very young, research on collectives has resulted in successes both in understanding and designing such systems. It is expected that as it matures and draws upon other disciplines related to collectives, this field will greatly expand the range of computationally addressable tasks. Moreover, in addition to drawing on them, such a fully developed field of collective intelligence may provide insight into already established scientific fields, such as mechanism design, economics, game theory, and population biology. This chapter provides a survey to the emerging science of collectives.

1.1 Just What is a "Collective"?

As computing power increases, becomes cheaper and is packed into smaller and smaller units, a new computational paradigm, one based on adaptive distributed computing is emerging. Whether used for control or optimization of complex engineered systems, or the analysis of natural systems, this new paradigm offers new and exciting solutions to the problems of the twenty first century. However, before the full strength of this powerful computational paradigm can be harnessed, some fundamental issues need to be addressed.

In this chapter we provide a survey of approaches to large distributed

systems called **collectives**. A collective is a large system of agents[1], where each agent has a **private utility** function it is trying to optimize adaptive utility-maximizing algorithms, called "agents", along a **world utility** function that measures the full system's performance[2]. Though system that meet this definition have been investigated in various field, no current discipline provides a general framework with which to design and study collectives.

Mechanism design, a subfield of economics, is perhaps the closest field addressing the "design" question posed in a collective [82, 87]. Mechanism design aims at finding the right "market mechanism" that will induce a set of agents to act in a manner specified by the system designer. Though this seems like a close match for what we expect a collective to achieve, conventional mechanism design is specifically designed for human agents and therefore is not meant to deal with arbitrary private and world utilities. Also, some issues essential to collectives (e.g., learning in agents) do not play a central role in it (see Section 1.2.3 for details).

Game theory, on the other hand, provides a good basis for the analysis of collectives [11, 19, 30, 87]. However, the principal focus of game theory is on the equilibrium behavior of fully rational agents. Unfortunately, large adaptive real world systems seldom operate at (or near) equilibrium, and due to the uncertainty in the agents' decision making, are rarely composed of fully rational agents. Furthermore practical issues fundamental to collectives (e.g., scaling) are not generally addressed in game theory (see Section 1.2.2 for details).

In the computer science domain, Reinforcement Learning (RL) [123, 221] and in particular, reinforcement learning in a Multi-Agent System (MAS) [53, 56, 112, 192] addresses the question of how in a large dynamic environment, one can learn to take actions to optimize a reward function. In general however, RL in a MAS does not address how the reward functions have to be crafted so that agents collectively act to optimize a world utility is not addressed. As a consequence, in traditional RL approach to multi-agent systems, each agent receives the full world reward as its private utility. Though this "solution" bypasses the incentive compatibility issue, it ignores the scalability issue. As such, though such systems work well where there are a small number of agents [56], they do not scale to system with hundreds or thousands of agents (see Section 1.2.1 for details).

Though mechanism design, game theory and reinforcement learning in multi-agent systems provide some of the ingredients required for a full fledged field of collectives, they fall short of providing a suitable starting point for the development of such a field. Furthermore, merging concepts

[1] we use the term "agent" to refer to the components of the system, though the various fields surveyed use different terminology (i.e., player in game theory)

[2] The world utility can be provided as part of the specifications of the system, or "constructed" by the designer, as discussed below.

from one of these fields to another is in general cumbersome due to the various assumptions – rarely explicit – deeply rooted in each field. What is needed for the field of collectives to develop and mature is a common language describing the various properties of collectives, a set of desirable properties, a theoretical framework, and a set of problems that will provide good testing grounds for new ideas in this field.

1.1.1 Distinguishing Characteristics of Collectives

Collectives can be characterized through many different distinguishing characteristics. In design problems there are many decisions (either explicit or implicit) that greatly affect the type of collective with which one ends up. Similarly, there are many decisions that determine what types of problems can be analyzed as collectives.

Since the chapters in this volume will focus on various design and analysis aspects of collectives, we briefly synopsize some distinguishing characteristics of collectives. These include the presence/absence of a well-defined world utility function; the forward/inverse approach; the presence/need for centralized control and/or communications; the presence/absence of a model; and scalability/robustness/adaptivity.

World Utility Function

Having a well-defined world utility function that concerns the behavior of the entire distributed system is crucial in the study of collectives. Such a world utility function provides an objective quantification of how well the system is performing. In that light, in a collective, we are not concerned with an unquantifiable "emergent" behavior of the system. Rather we are interested in how the system meets the pre-specified world utility (of course, nothing precludes the world utility from depending on the emergent behavior of the system, assuming such behavior can be quantified).

The most natural type of world utility is a *provided* utility, one that comes as part of the problem definition and specifies the overall performance criteria that the collectives needs to meet. Examples of such world utilities include total throughput in a data network, total scientific information gathered by a team of deployables, total information downloaded by a constellation of satellites, the valuation of a company, or the percentage of available free energy exploited by an ecosystem.

However, the lack of a provided world utility does not preclude a collective-based approach to a problem. In such a case, assuming the agents have some utility functions associated with them, a world utility can be constructed (e.g., construct a social welfare function in economics). Examples of such world utilities include sum of agent utilities, sum of agent utilities and variances, and the utility of the worst-off agent. Note that optimizing each of these *constructed* world utilities would result in different system behav-

ior. What is particularly interesting in such problems is the relationship between the agents' initial utility functions and the utility functions that they ought to pursue in order to optimize the constructed world utility function.

Forward (Analysis) vs. Inverse (Design) Problem

Whether it has a provided or constructed world utility, a collective can be approached from two very different perspectives. Analysis or the forward problem, and design or the inverse problem.

The **forward problem** focuses on how the localized attributes of a collective induce global behavior and thereby determine system performance. Generally, this problem arises in the study of already existing complex systems, and is most naturally applicable to biological systems, or systems that can be viewed as such. Examples of such systems include ecosystems, or a living cell, where in each case, the local interactions (species and organelles, respectively) lead to complex emergent behavior at a large scale.

Engineered systems such as processes (e.g., the space shuttle maintenance and refurbishment process) or (economic) organizations can also be viewed as forward problems in collectives. In those cases, the analysis approach can lead to predictive models and detect interactions among components of the system that may lead to breakdowns (e.g., determining whether a component considered "safe" can cause a critical malfunction when it is put in interaction with another "safe" component).

The **inverse problem** on the other hand, arises when we wish to design a system to induce behavior which optimizes the world utility. Here, the designer either has the freedom to assign the private utility functions of the agents (e.g., determine what each satellite or router should be doing) or needs to design incentives that will be added to the pre-existing private utilities of the agents (e.g., economics, where agents are humans). In either case though, the focus is on guiding towards states where the world utility is high.

Centralized communication or control

Though not in the formal definition of a collective, many collectives are decentralized systems. With few exceptions, it will be difficult, if not impossible, to have centralized control in a collective, not only because reaching each agent may be problematic, but more fundamentally, because in many cases a centralized algorithm may not be able to determine what each agent should do.

Similarly, though some amount of global communication (e.g., broadcasting) may be possible, in general there will be little to no centralized communication, where a small subset of agents not only communicates with all the other agents, but communicates differently with each one of those other agents. Establishing the amount of allowed (or possible) cen-

tralized communication and control will be one of the fundamental issues in a collective.

Model-Based vs. Model-Free

Another important characteristic of a collective is the presence/absence of a model describing the dynamics of the system. A **model-based** approaches consist of:

1. Constructing a detailed model of the dynamics governing the collective;
2. Learning the function which maps the parameters of the model to the resulting dynamics of the system (in practice, this step can involve significant hand-tuning); and
3. a) Drawing conclusions about this system based on the model (forward problem);
   b) Determining parameters of the model that will yield desired behavior (inverse problem).

A fundamentally different approach however, is to dispense with building a model altogether, on the grounds that large, complex systems are generally noisy, faulty, and often operate in non-stationary environments. In such cases, coming up with a detailed model that captures the dynamics in an accurate manner is often extraordinarily difficult.

A **model-free** approach hand relies on the agents "reacting" to the environment (e.g., through a reinforcement learning mechanism). As such they avoid explicitly modeling the system in which they operate, and in particular, avoid the potentially infinite regress when one agent tries to model another's behavior and that other agent is itself modeling the first agent's behavior.

The model-based vs. model-free choice has significant consequences in how the system can adapt, scale up, and how lessons learned from one domain can map to another one. A model-based approach may be the choice for domains where the designers can develop detailed models and have a moderate degree of control over the environment. However, in domains where detailed models are not available, or where there is reason to believe changes in the environment can lead to significant deviations from any model, a model-free approach is preferable.

Scalability

One of the implicit defining properties of a collective is that it is a *large* system of distributed agents. As such, scalability is a fundamental property of any approach that aims to study/design a collective. Though this does not preclude extending extant analysis/design tools appropriate for single

(or small) systems to large systems, it does suggest that in most instances, new ways of approaching the problem are likely to be more appropriate (e.g., a game theoretic equilibrium analysis for a million nano-devices is unlikely to provide useful insight into the behavior of the collective.)

Adaptivity

Though scalability does not require that the system be adaptive, it provides a strong impetus to move in that direction. Any approach that allows adaptivity, or learning, will have a significant advantage over one that does not, simply because the larger a system, the more difficult it will be to know a priori all the "right moves" for each agent.

Furthermore, the need for adaptivity extends beyond each agent in the collective. Indeed the structure of the collective itself (e.g., the communication channels among the agents, the agents' utility function) in many cases is adaptive. In natural collectives this system-level adaptivity is generally implicit (e.g., the interaction among species in an ecosystem or the relationship among employees in a company), whereas in artificial systems it must be built in.

Robustness

Another desirable property of a collective is that it be robust, i.e., that the collective not require that many details (e.g., parameters) be set just right it to perform well. Clearly, as the number of agents in a collective goes up, it will become increasingly difficult to ensure failure-free operation of each agent. It is therefore imperative that the structure of the collective be insensitive to the specific operation of a small subset of its agents (e.g., in general the poor performance of one employee does not bring a company down, or the demise of a single individual does not result in the extinction of a species).

1.1.2 Canonical Experimental Domains

The previous section provided a list of distinguising characteristics of collectives. The usefulness of these characteristics is in their providing a common language for a field of collectives. For example, a particular instance of data routing in a telecommunications network can be characterized as "a model-free inverse problem involving a provided world utility function where there is limited broadcast information but no form of global control."

We now provide examples of both engineered and natural systems which are ideally suited to be studied as collectives. For each, we provide one or more world utility functions, discuss how it can be approached (e.g., forward/inverse problem), and what assumptions (e.g., is it model-based?) and restrictions (e.g., is global communication possible?) are present.

- *Control system for constellations of communication satellites:* A candidate world utility for this problem is a measure of (potentially importance weighted) information transferred. It is an example of an inverse problem, where centralized communication or control is likely to be difficult or impossible due to physical constraints (e.g., time lag), and where a model of the data flow is likely to be inadequate.

- *Control system for constellations of planetary exploration vehicles:* A potential world utility for such a problem is a measure of the quality of scientific data collected. Though this can be viewed as an example of an inverse design problem (as with constellations of satellites), it can also be approached as a forward problem, particularly if the vehicles have have characteristics which cannot be altered (e.g., vehicles are built and we are confronted with the problem of predicting the behavior of the collective).

- *Control system for routing over a communication network:* An obvious world utility for this problem is the total throughput of the communication network. Centralized communication or control in such a network is all but impossible, but some amount of broadcast information can filter its way to all the agents at regular time intervals. As an inverse problem, one would be required to design the private utility functions of the agents. As a forward problem on an already functioning network, one could determine the stress points of the system, or the states which would cause the largest congestions in the network.

- *Air Space Management:* Given a problem specification where there is some leeway in modifying the course and speed of airplanes, a potential world utility is minimizing delays at airports. The system designers are faced with the inverse problem of determining the incentives for the agents (whether they be pilots or air traffic controllers) so that their behavior (e.g., arrival times to the airport's airspace) optimizes the world utility. This is a case where though global communication is possible, global control is not.

- *Managing a power grid:* A world utility based on the efficiency of the grid would be a good starting point for an inverse problem, involving some degree of centralized communication or control. An alternative world utility may be robustness. In such a case a forward problem would involve finding how quickly the system responds to certain disturbances, and how the system interactions can be modified so as to limit the propagation of those disturbances.

- *Job scheduling across a computational grid:* A candidate world utility is the efficiency in processing the jobs entering the system. This problem is very similar to managing a power grid, but provides a glimpse

at the inverse problem: how should one set the rewards of the computational nodes so that they process the most number of jobs collectively? A model-free solution involving learners at the computational nodes would be based on limited global communication.

- *Control of the elements of a nanocomputer:* A potential world utility for this problem is how well certain computations are carried out by the nanocomputer. In an inverse problem, one would focus on determining the structure of the adaptive system which would lead the agents to perform the desired computations. A particular instance of an inverse problem of this nature is the selection of subsets of faulty devices, where the world utility is total aggregate error of the selected devices.

- *Study of a protocell:* A potential world utility for this problem is the length of time the protocell maintains its functionality. As a forward problem, this problem consists of modeling the behavior of the system based on the organelles and their functions/interactions. With more leeway in the definition of the functions the organelles perform, one can view this as an interesting inverse problem: What should the organelles try to achieve to maintain the structure and functionality of the protocell?

- *Study/Design of an ecosystem:* One world utility for the study of an ecosystem is the total bio-mass of the ecosystem. In a model-based forward problem, one can study the effect of various interactions on the world utility. Alternatively, as an inverse problem, one can investigate how to design an ecosystem which will provide the best sustainable bio-diversity for a given mass (e.g., for a long term space mission).

- *Design of incentives in a Company:* A "simple" world world utility for a company is the valuation of the company (share price times the number of outstanding shares). The inverse problem consists of determining how to design incentives that will induce the companies valuation to go up (e.g., what set of salaries/benefits/stock options will induce the employees to take actions that will benefit the corporation).

All of these problem share the property that they are inherently distributed systems where the interactions among the agents leads to complex behavior. Though each one can be approached by conventional methods, how those methods need to be modified to suit the particular application will be different in each case. The aim of this chapter is to both accentuate the similarities among these problems and also to highlight the need for a general approach which would address all these problems within the same framework.

1.2 Review of Literature Related to Collectives

There are many approaches to analyzing and designing collectives that do not exactly meet the needs of a "field of collectives" yet provide some part of the equation. The rest of this section consists of brief presentations of some of these approaches, and in particular characterizes them in terms of the properties of collectives discussed above.

1.2.1 AI and Machine Learning

There is an extensive body of work in AI and machine learning that is related to the design of collectives. Indeed, one of the most famous speculative works in the field can be viewed as an argument that AI should be approached as a design of collectives problem [163]. Below, we discuss some topics relevant to collectives from this domain.

Distributed Artificial Intelligence

The field of Distributed Artificial Intelligence (DAI) has arisen as more and more traditional Artificial Intelligence (AI) tasks have migrated toward parallel implementation. The most direct approach to such implementations is to directly parallelize AI production systems or the underlying programming languages [79, 189]. An alternative and more challenging approach is to use distributed computing, where not only are the individual reasoning, planning and scheduling AI tasks parallelized, but there are *different modules* with different such tasks, concurrently working toward a common goal [118, 119, 143].

In a DAI, one needs to ensure that the task has been modularized in a way that improves efficiency. Unfortunately, this usually requires a central controller whose purpose is to allocate tasks and process the associated results. Moreover, designing that controller in a traditional AI fashion often results in brittle solutions. Accordingly, recently there has been a move toward both more autonomous modules and fewer restrictions on the interactions among the modules [194].

Despite this evolution, DAI maintains the traditional AI concern with a pre-fixed set of *particular* aspects of intelligent behavior (*e.g.* reasoning, understanding, learning etc.) rather than on their *cumulative* character. As the idea that intelligence may have more to do with the interaction among components started to take shape [41, 42], focus shifted to concepts (*e.g.*, multi-agent systems) that better incorporated that idea [121].

Multi-Agent Systems

The field of Multi-Agent Systems (MAS) is concerned with the interactions among the members of such a set of agents [40, 92, 121, 204, 222], as well as the inner workings of each agent in such a set (*e.g.*, their learning

algorithms) [36, 37, 38]. As in computational ecologies and computational markets (see below), a well-designed MAS is one that achieves a global task through the actions of its components. The associated design steps involve [121]:

1. Decomposing a global task into distributable subcomponents, yielding tractable tasks for each agent;

2. Establishing communication channels that provide sufficient information to each of the agents for it to achieve its task, but are not too unwieldly for the overall system to sustain; and

3. Coordinating the agents in a way that ensures that they cooperate on the global task, or at the very least does not allow them to pursue conflicting strategies in trying to achieve their tasks.

Step (3) is rarely trivial; one of the main difficulties encountered in MAS design is that agents act selfishly and artificial cooperation structures have to be imposed on their behavior to enforce cooperation [13]. An active area of research, which holds promise for addressing parts the design of collectives problem, is to determine how selfish agents' "incentives" have to be engineered in order to avoid problems such as the tragedy of the commons (TOC) [209]. (This work draws on the economics literature, which we review separately below.) When simply providing the right incentives is not sufficient, one can resort to strategies that actively induce agents to cooperate rather than act selfishly. In such cases coordination [205], negotiations [135], coalition formation [193, 195, 249] or contracting [3] among agents may be needed to ensure that they do not work at cross purposes.

Unfortunately, all of these approaches share with DAI and its offshoots the problem of relying on hand-tailoring, and therefore being difficult to scale and often nonrobust. In addition, except as noted in the next subsection, they involve little to no adaptivity, and therefore the constituent computational elements are usually not as robust as they would need to be to provide the foundation for the field of collectives.

Reinforcement Learning

The maturing field of Reinforcement Learning (RL) provides a much needed tool for the types of problems addressed by collectives. The goal of an RL algorithm is to determine how, using those reward signals, the agent should update its action policy to maximize its utility [123, 220, 221, 232]. Because RL generally provides model-free[3] and "online" learning features, it is ideally suited for the distributed environment where a "teacher" is not available and the agents need to learn successful strategies based on

[3]There exist some model-based variants of traditional RL. See for example [8].

"rewards" and "penalties" they receive from the overall system at various intervals. It is even possible for the learners to use those rewards to modify *how* they learn [199, 200].

Although work on RL dates back to Samuel's checker player [191], relatively recent theoretical [232] and empirical results [56, 224] have made RL one of the most active areas in machine learning. Many problems ranging from controlling a robot's gait to controlling a chemical plant to allocating constrained resource have been addressed with considerable success using RL [97, 114, 166, 186, 247]. In particular, the RL algorithms $TD(\lambda)$ (which rates potential states based on a *value function*) [220] and Q–learning (which rates action-state pairs) [232] have been investigated extensively. A detailed investigation of RL is available in [123, 221, 232].

Intuitively, one might hope that RL would help us solve the distributed control problem, since RL is adaptive, and, in general mode-free. However, by itself, conventional single-agent RL does not provide a means for controlling large, distributed systems. The problem is that the space of possible action policies for such systems is too big to be searched. So although powerful and widely applicable, solitary RL algorithms will not generally perform well on large distributed heterogeneous problems. It is however natural to consider deploying many RL algorithms rather than a single one for these large distributed problems.

Reinforcement Learning-Based Multi-Agent Systems

Because it neither requires explicit modeling of the environment nor having a "teacher" that provides the "correct" actions, the approach of having the individual agents in a MAS use RL is well-suited for MAS's deployed in domains where one has little knowledge about the environment and/or other agents. There are two main approaches to designing such MAS's:
(i) One has 'solipsistic agents' that don't know about each other and whose RL rewards are given by the performance of the entire system (so the joint actions of all other agents form an "inanimate background" contributing to the reward signal each agent receives);
(ii) One has 'social agents' that explicitly model each other and take each others' actions into account.
Both (i) and (ii) can be viewed as ways to (try to) coordinate the agents in a MAS in a robust fashion.

Solipsistic Agents: MAS's with solipsistic agents have been successfully applied to a multitude of problems [56, 96, 107, 192, 198]. However, scaling to large systems is a major issue with solipsistic agents. The problem is that each agent must be able to discern the effect of its actions on the overall performance of the system, since that performance constitutes its reward signal. As the number of agents increases though, the effects of any one agent's actions (signal) will be swamped by the effects of other agents (noise), making the agent unable to learn well, if at all. In addition, of

course, solipsistic agents cannot be used in situations lacking centralized calculation and broadcast of the single global reward signal.

Social agents: MAS's whose agents take the actions of other agents into account synthesize RL with game theoretic concepts (*e.g.*, Nash equilibrium). They do this to try to ensure that the overall system both moves toward achieving the overall global goal and avoids often deleterious oscillatory behavior [53, 85, 111, 113, 112]. To that end, the agents incorporate internal mechanisms that actively model the behavior of other agents. In general this approach involves hand-tailoring for the problem, and there are some well-studied domains (El Farol Bar problem) in which such modeling is self-defeating [5, 238].

1.2.2 Game Theory

Game theory is the branch of mathematics concerned with formalized versions of "games", in the sense of chess, poker, nuclear arms races, and the like [11, 19, 30, 73, 87, 148, 66, 207]. It is perhaps easiest to describe it by loosely defining some of its terminology, which we do here and in the next subsection.

The simplest form of a game is that of 'non-cooperative single-stage extensive-form' game, which involves the following situation: There are two or more agents (called 'players' in the literature), each of which has a pre-specified set of possible actions that it can follow. (A 'finite' game has finite sets of possible actions for all the players.) In addition, each agent i has a utility function (also called a 'payoff matrix' for finite games). This maps any 'profile' of the action choices of all agents to an associated utility value for agent i. (In a 'zero-sum' game, for every profile, the sum of the payoffs to all the agents is zero.)

The agents choose their actions in a sequence, one after the other. The structure determining what each agent knows concerning the action choices of the preceding agents is known as the 'information set'.[4] Games in which each agent knows exactly what the preceding ('leader') agent did are known as 'Stackelberg games'.

In a 'multi-stage' game, after all the agents choose their first action, each agent is provided some information concerning what the other agents did. The agent uses this information to choose its next action. In the usual formulation, each agent gets its payoff at the end of all of the game's stages.

An agent's 'strategy' is the rule it elects to follow mapping the information it has at each stage of a game to its associated action. It is a 'pure strategy' if it is a deterministic rule. If instead the agent's action is chosen

[4] While stochastic choices of actions is central to game theory, most of the work in the field assumes the information in information sets is in the form of definite facts, rather than a probability distribution. Accordingly, there has been relatively little work incorporating Shannon information theory into the analysis of information sets.

by randomly sampling from a distribution, that distribution is known a 'mixed strategy'. Note that an agent's strategy concerns *all* possible sequences of provided information, even any that cannot arise due to the strategies of the other agents.

Any multi-stage extensive-form game can be converted into a 'normal form' game, which is a single-stage game in which each agent is ignorant of the actions of the other agents, so that all agents choose their actions "simultaneously". This conversion is achieved by having the "actions" of each agent in the normal form game correspond to an entire strategy in the associated multi-stage extensive-form game. The payoffs to all the agents in the normal form game for a particular strategy profile is then given by the associated payoff matrices of the multi-stage extensive form-game.

Nash Equilibrium

A 'solution' to a game, or an 'equilibrium', is a profile in which every agent behaves "rationally". This means that every agent's choice of strategy optimizes its utility subject to a pre-specified set of conditions. In conventional game theory those conditions involve, at a minimum, perfect knowledge of the payoff matrices of all other players, and often also involve specification of what strategies the other agents adopted and the like. In particular, a 'Nash equilibrium' is a a profile where each agent has chosen the best strategy it can, *given the choices of the other agents.* A game may have no Nash equilibria, one equilibrium, or many equilibria in the space of pure strategies. A beautiful and seminal theorem due to Nash proves that every game has at least one Nash equilibrium in the space of mixed strategies [171].

There are several different reasons one might expect a game to result in a Nash equilibrium. One is that it is the point that perfectly rational Bayesian agents would adopt, assuming the probability distributions they used to calculate expected payoffs were consistent with one another [10, 124]. A related reason, arising even in a non-Bayesian setting, is that a Nash equilibrium provides "consistent" predictions, in that if all parties predict that the game will converge to a Nash equilibrium, no one will benefit by changing strategies. Having a consistent prediction does not ensure that all agents' payoffs are maximized though. The study of small perturbations around Nash equilibria from a stochastic dynamics perspective is just one example of a 'refinement' of Nash equilibrium, that is a criterion for selecting a single equilibrium state when more than one is present [154].

Cooperative Game Theory

In cooperative game theory the agents are able to enter binding contracts with one another, and thereby coordinate their strategies. This allows the agents to avoid being "stuck" in Nash equilibria that are Pareto inefficient, that is being stuck at equilibrium profiles in which all agents would benefit

if only they could agree to all adopt different strategies, with no possibility of betrayal. The *characteristic function* of a game involves subsets ('coalitions') of agents playing the game. For each such subset, it gives the sum of the payoffs of the agents in that subset that those agents can guarantee if they coordinate their strategies. An *imputation* is a division of such a guaranteed sum among the members of the coalition. It is often the case that for a subset of the agents in a coalition one imputation *dominates* another, meaning that under threat of leaving the coalition that subset of agents can demand the first imputation rather than the second. So the problem each agent i is confronted with in a cooperative game is which set of other agents to form a coalition with, given the characteristic function of the game and the associated imputations i can demand of its partners. There are several different kinds of solution for cooperative games that have received detailed study, varying in how the agents address this problem of who to form a coalition with. Some of the more popular are the 'core', the 'Shapley value', the 'stable set solution', and the 'nucleolus'.

In the real world, the actual underlying game the agents are playing does not only involve the actions considered in cooperative game theory's analysis of coalitions and imputations. The strategies of that underlying game also involve bargaining behavior, considerations of trying to cheat on a given contract, bluffing and threats, and the like. In many respects, by concentrating on solutions for coalition formation and their relation with the characteristic function, cooperative game theory abstracts away these details of the true underlying game. Conversely though, progress has recently been made in understanding how cooperative games can arise from non-cooperative games, as they must in the real world [11].

Evolution and Learning in Games

Not surprisingly, game theory has come to play a large role in the field of multi-agent systems. In addition, due to Darwinian natural selection, one might expect game theory to be quite important in population biology, in which the "utility functions" of the individual agents can be taken to be their reproductive fitness. There is an entire subfield of game theory concerned with this connection with population biology, called 'evolutionary game theory' [155, 157].

To introduce evolutionary game theory, consider a game in which all players share the same space of possible strategies, and there is an additional space of possible 'attribute vectors' that characterize an agent, along with a probability distribution g across that new space. (Examples of attributes in the physical world could be things like size, speed, etc.) We select a set of agents to play a game by randomly sampling g. Those agents' attribute vectors jointly determine the payoff matrices of each of the individual agents. (Intuitively, what benefit accrues to an agent for taking a particular action depends on its attributes and those of the other agents.)

However each agent i has limited information concerning both its attribute vector and that of the other players in the game, information encapsulated in an 'information structure'. The information structure specifies how much each agent knows concerning the game it is playing.

In this context, we enlarge the meaning of the term "strategy" to not just be a mapping from information sets and the like to actions, but from entire information structures to actions. In addition to the distribution g over attribute vectors, we also have a distribution over strategies, h. A strategy s is a 'population strategy' if h is a delta function about s. Intuitively, we have a population strategy when each animal in a population "follows the same behavioral rules", rules that take as input what the animal is able to discern about its strengths and weakness relative to those other members of the population, and produce as output how the animal will act in the presence of such animals.

Given g, a population strategy centered about s, and its own attribute vector, any player i in the support of g has an expected payoff for any strategy it might adopt. When i's payoff could not improve if it were to adopt any strategy other than s, we say that s is 'evolutionary stable'. Intuitively, an evolutionary stable strategy is one that is stable with respect to the introduction of mutants into the population.

Now consider a sequence of such evolutionary games. Interpret the payoff that any agent receives after being involved in such a game as the 'reproductive fitness' of that agent, in the biological sense. So the higher the payoff the agent receives, in comparison to the fitnesses of the other agents, the more "offspring" it has that get propagated to the next game. In the continuum-time limit, where games are indexed by the real number t, this can be formalized by a differential equation. This equation specifies the derivative of g_t evaluated for each agent i's attribute vector, as a monotonically increasing function of the relative difference between the payoff of i and the average payoff of all the agents. (We also have such an equation for h.) The resulting dynamics is known as 'replicator dynamics', with an evolutionary stable population strategy, if it exists, being one particular fixed point of the dynamics.

Now consider removing the reproductive aspect of evolutionary game theory, and instead have each agent propagate to the next game, with "memory" of the events of the preceding game. Furthermore, allow each agent to modify its strategy from one game to the next by "learning" from its memory of past games, in a bounded rational manner. The field of learning in games is concerned with exactly such situations [86, 12, 17, 26, 70, 126, 178, 173]. Most of the formal work in this field involves simple models for the learning process of the agents. For example, in 'fictitious play' [86], in each successive game, each agent i adopts what would be its best strategy if its opponents chose their strategies according to the empirical frequency distribution of such strategies that i has encountered in the past. More sophisticated versions of this work employ simple Bayesian

learning algorithms, or re-inventions of some of the techniques of the RL community [190]. Typically in learning in games one defines a payoff to the agent for a sequence of games, for example as a discounted sum of the payoffs in each of the constituent games. Within this framework one can study the long term effects of strategies such as cooperation and see if they arise naturally and if so, under what circumstances.

Many aspects of real world games that do not occur very naturally otherwise arise spontaneously in these kinds of games. For example, when the number of games to be played is not pre-fixed, it may behoove a particular agent i to treat its opponent better than it would otherwise, since i *may* have to rely on that other agent's treating it well in the future, if they end up playing each other again. This framework also allows us to investigate the dependence of evolving strategies on the amount of information available to the agents [159]; the effect of communication on the evolution of cooperation [160, 162]; and the parallels between auctions and economic theory [108, 161].

In many respects, learning in games is even more relevant to the study of collectives than is traditional game theory. However in general, it lacks a well defined world utility and is almost exclusively focused on the forward problem, making it a difficult starting point for a field of collectives.

1.2.3 Other Social Science–Inspired Systems

Some human economies provides examples of naturally occurring systems that can be viewed as a (more or less) well-performing collectives. The field of economics provides much more though. Both empirical economics (*e.g.*, economic history, experimental economics) and theoretical economics (*e.g.*, general equilibrium theory [4], theory of optimal taxation [164]) provide a rich literature on strategic situations where many parties interact. In fact, much of economics can be viewed as concerning how to maximize certain constrained kinds of world utilities, when there are certain (very strong) restrictions on the individual agents and their interactions, and in particular when we have limited freedom in setting the utility functions of those agents.

Mechanism Design

One way to try to induce a large collective to reach an equilibrium point without centralize control is via an auction.[5] (This is the approach usu-

[5] We do not discuss general equilibrium theory here in detail, because though it deals with the interaction among multiple markets to set the market "clearing" price for the goods, it is not appropriate for the study of collectives: it requires centralized control (Walrasian auctioner), does not allow for dynamic interactions and in general, there is no reason to believe that having the markets clear optimizes a world utility.

ally employed in computational markets — see below.) Along with optimal taxation and public good theory [137], the design of auctions is the subject of the field of mechanism design. Broadly defined, mechanism design is concerned with the incentives that must be applied to any set of agents that interact and exchange goods [87, 164, 229] in order to get those agents to exhibit desired behavior. Usually the desired behavior concerns prespecified 'inherent' utility functions of some sort for each of the individual agents. In particular, mechanism design is often concerned with the incentives that must be superimposed on such inherent utility functions to guide the agents to a '(Pareto) efficient' (or 'Pareto optimal') point, that is to a point in which no agent's inherent utility can be improved without hurting another agent's inherent utility [86, 87].

One particularly important type of such an incentive scheme is an auction. When many agents interact in a common environment often there needs to be a structure that supports the exchange of goods or information among those agents. Auctions provide one such (centralized) structure for managing exchanges of goods. For example, in the English auction all the agents come together and 'bid' for a good, and the price of the good is increased until only one bidder remains, who gets the good in exchange for the resource bid. As another example, in the Dutch auction the price of a good is decreased until one buyer is willing to pay the current price.

All auctions perform the same task: match supply and demand. As such, auctions are one of the ways in which price equilibration among a set of interacting agents can be achieved. However very few world utilities have their maximum occur at a point that is Pareto optimal for the pre-set inherent utility functions. Accordingly, unless we are very fortunate in the relation between those inherent utility functions and (in general separately specified) world utility, knowing how to induce such a Pareto optimal point is of little value. For example, in a transaction in an English auction both the seller and the buyer benefit. They may even have arrived at an allocation which is efficient. However, in that the winner may well have been willing to pay more for the good, such an outcome may confound the goal of the market designer, if that designer's goal is to maximize revenue. This point is returned to below, in the context of computational economics.

Another, perhaps more intuitive perspective, is to view the restrictions of mechanism design as concerning the private utility functions of the individual agents. Typically in mechanism design the private utility function for each agent η, which maps states of the entire world (including the internal state of the agent itself) to $\mathcal{R}$, is of the form $\gamma_\eta(x_{\eta,1}, x_{\eta,2}, ..., x_{\eta,n}, T_\eta(y_{\eta,1}, y_{\eta,2}, ..., y_{\eta,m}))$, where $\gamma_\eta(.)$ is agent η's pre-fixed inherent utility function, the $x_{\eta,1}, x_{\eta,2}, ..., x_{\eta,n}$ constitute the first n of the $n + k$ variables that that function depends on, and $T_\eta(.)$ is the $\mathcal{R}^k$-valued "mechanism" function the designer can set, the $y_{\eta,1}, y_{\eta,2}, ..., y_{\eta,m}$ being the variables making up its arguments. Unlike the private utility world utility can depend on all of the $x_{\eta,1}, ..., x_{\eta,n}, y_{\eta,1}, y_{\eta,2}, ..., y_{\eta,m}$ directly (as well as depend on other

entirely different variables). As an example, the $y_{\eta,1}, y_{\eta,2}, ..., y_{\eta,m}$ could be a set of all agents' bids at an auction, $T_\eta(.)$ could be $\mathcal{R}^2$-valued, giving the amount of change in η's owned quantities of both money and the item up for bid, and the $x_{\eta,1}, ..., x_{\eta,n}$ could parameterize η's happiness trade-off relating owned quantities of the good and of money.

Typically $\gamma_\eta(.)$ and the choice of what variables make up the arguments $y_{\eta,1}, y_{\eta,2}, ..., y_{\eta,m}$ to T_η are fixed *a priori*, with only the function $T_\eta(.)$ allowed to vary in the design. In addition, often there are *a priori* restrictions on the functional form of the T_η. For example, often the T_η are not allowed to vary with η. More precisely, usually they must be invariant under the transformation $\eta \rightarrow \eta'$ in both the index to the function and the indices to its arguments. This means in particular that the designer can't "cheat" and have the functional forms of the T_η vary from one η to another in a way that reflects the variations across the (often pre-determined) associated vectors $(x_{\eta,1}, ..., x_{\eta,n})$. For example, typically an auction mechanism determines who gets what goods for what price in a manner that is independent of the identities of the bidding agents, and in particular does not directly reflect any internal happiness trade-off parameters of the agents that aren't reflected in their bids.

From the perspective of a collective, these kinds of restrictions on private utilities only hold in a small subset of the potential computational problems, and constitute a severe handicap in other scenarios. Another limitation of most of the work on mechanism design is that either it assumes a particular computational model for the agent, or (more commonly) focuses on (game-theoretic) equilibria. This limited nature of the treatment of off-equilibrium scenarios is intimately related to the restrictions on the form of the private utility. If there are no restrictions on the private utilities, then there is a trivial solution for how to set such utilities to maximize the world utility at equilibrium: Have each such utility simply equal the world utility, in a so-called "team game". To have the analysis be non-trivial, restrictions like those on the private utilities are needed.

In practice though, no real system is at a game-theoretic equilibrium, due to bounded rationality. In particular, it means that if one considers mechanism design in the limiting case of no restrictions on $\gamma(.)$, the associated "mechanism design solution" of a team game often will result in poor performance [238]. Team theory [105, 153] is one approach that has been tried to circumvent this problem. The idea there is to remove all notions of a private or inherent utility, and solve directly for the strategy profile that will maximize the world utility. Needless to say though, such an approach becomes extraordinarily difficult for all but the simplest problems, and requires centralized, completely personalized control and communication, and exact modeling of the system's dynamics.

Computational Economics

'Computational economies' are schemes inspired by economics, and more specifically by general equilibrium theory and mechanism design theory, for managing the components of a distributed computational system. They work by having a 'computational market', akin to an auction, guide the interactions among those components. Such a market is defined as any structure that allows the components of the system to exchange information on relative valuation of resources (as in an auction), establish equilibrium states (*e.g.*, determine market clearing prices) and exchange resources (*i.e.*, engage in trades).

Such computational economies can be used to investigate real economies and biological systems [31, 34, 35, 128]. They can also be used to design distributed computational systems. For example, such computational economies are well-suited to some distributed resource allocation problems, where each component of the system can either directly produce the "goods" it needs or acquire them through trades with other components. Computational markets often allow for far more heterogeneity in the components than do conventional resource allocation schemes. Furthermore, there is both theoretical and empirical evidence suggesting that such markets are often able to settle to equilibrium states. For example, auctions find prices that satisfy both the seller and the buyer which results in an increase in the utility of both (else one or the other would not have agreed to the sale). Assuming that all parties are free to pursue trading opportunities, such mechanisms move the system to a point where all possible bilateral trades that could improve the utility of both parties are exhausted.

Now restrict attention to the case, implicit in much of computational market work, with the following characteristics: First, world utility can be expressed as a monotonically increasing function F where each argument i of F can in turn be interpreted as the value of a pre-specified utility function f_i for agent i. Second, each of those f_i is a function of an i-indexed 'goods vector' x_i of the non-perishable goods "owned" by agent i. The components of that vector are $x_{i,j}$, and the overall system dynamics is restricted to conserve the vector $\sum_i x_{i,j}$. (There are also some other, more technical conditions.) As an example, the resource allocation problem can be viewed as concerning such vectors of "owned" goods.

Due to the second of our two conditions, one can integrate a market-clearing mechanism into any system of this sort. Due to the first condition, since in a market equilibrium with non-perishable goods no (rational) agent ends up with a value of its utility function lower than the one it started with, the value of the world utility function must be higher at equilibrium than it was initially. In fact, so long as the individual agents are smart enough to avoid all trades in which they do not benefit, any computational market can only improve this kind of world utility, even if it does not achieve the market equilibrium.

This line of reasoning provides one of the main reasons to use computational markets in those situations in which they can be applied. Conversely, it underscores one of the major limitations of such markets: Starting with an arbitrary world utility function with arbitrary dynamical restrictions, it may be quite difficult to cast that function as a monotonically increasing F taking as arguments a set of agents' goods-vector-based utilities f_i, if we require that those f_i be well-enough behaved that we can reasonably expect the agents to optimize them in a market setting.

One example of a computational economy being used for resource allocation is Huberman and Clearwater's use of a double–blind auction to solve the complex task of controlling the temperature of a building. In this case, each agent (individual temperature controller) bids to buy or sell cool or warm air. This market mechanism leads to an equitable temperature distribution in the system [116]. Other domains where market mechanisms were successfully applied include purchasing memory in an operating systems [50], allocating virtual circuits [75], "stealing" unused CPU cycles in a network of computers [69, 230], predicting option futures in financial markets [185], and numerous scheduling and distributed resource allocation problems [138, 142, 210, 218, 234, 235].

Computational economics can also be used for tasks not tightly coupled to resource allocation. For example, following the work of Maes [151] and Ferber [74], Baum shows how by using computational markets a large number of agents can interact and cooperate to solve a variant of the blocks world problem [22, 23]. However, market-based computational economics relies on both centralized communication and centralized control to some degree, raising scalability issues. Furthermore, in practice, the applicability of computational economies depends greatly on the domain [225], making it a difficult starting point for a field of collectives.

1.2.4 Biologically Inspired Systems

Properly speaking, biological systems do not involve utility functions and searches across them with learning algorithms. However it has long been appreciated that there are many ways in which viewing biological systems as involving searches over such functions can lead to deeper understanding of them [203, 244]. Conversely, some have argued that the mechanism underlying biological systems can be used to help design search algorithms [109].[6]

These kinds of reasoning which relate utility functions and biological systems have traditionally focussed on the case of a single biological system operating in some external environment. If we extend this kind of reason-

[6]See [150, 236] though for some counter-arguments to the particular claims most commonly made in this regard.

ing, to a set of biological systems that are co-evolving with one another, then we have essentially arrived at biologically-based collectives. This section discusses some of how previous work in the literature bears on this relationship between collectives and biology.

Population Biology and Ecological Modeling

The fields of population biology and ecological modeling are concerned with the large-scale "emergent" processes that govern the systems that consist of many (relatively) simple entities interacting with one another [24, 99]. As usually cast, the "simple entities" are members of one or more species, and the interactions are some mathematical abstraction of the process of natural selection as it occurs in biological systems (involving processes like genetic reproduction of various sorts, genotype-phenotype mappings, inter and intra-species competitions for resources, etc.). Population Biology and ecological modeling in this context addresses questions concerning the dynamics of the resultant ecosystem, and in particular how its long-term behavior depends on the details of the interactions between the constituent entities. Broadly construed, the paradigm of ecological modeling can even be broadened to study how natural selection and self-regulating feedback creates a stable planet-wide ecological environment—Gaia [144].

The underlying mathematical models of other fields can often be usefully modified to apply to the kinds of systems population biology is interested in [14]. (See also the discussion in the game theory subsection above.) Conversely, the underlying mathematical models of population biology and ecological modeling can be applied to other non-biological systems. In particular, those models shed light on social issues such as the emergence of language or culture, warfare, and economic competition [71, 72, 88]. They also can be used to investigate more abstract issues concerning the behavior of large complex systems with many interacting components [89, 98, 156, 176, 184].

Going a bit further afield, an approach that is related in spirit to ecological modeling is 'computational ecologies'. These are large distributed systems where each component of the system's acting (seemingly) independently results in complex global behavior. Those components are viewed as constituting an "ecology" in an abstract sense (although much of the mathematics is not derived from the traditional field of ecological modeling). In particular, one can investigate how the dynamics of the ecology is influenced by the information available to each component and how cooperation and communication among the components affects that dynamics [115, 117].

Although in some ways the most closely related to collectives of the current ecology-inspired research, the fields of population biology and computational ecologies do not provide a full science of collectives. These fields are primarily concerned with the "forward problem" of determining the dynamics that arises from certain choices of the underlying system. Un-

less one's desired dynamics is sufficiently close to some dynamics that was previously catalogued (during one's investigation of the forward problem), one has very little information on how to set up the components and their interactions to achieve that desired dynamics.

Swarm Intelligence

The field of 'swarm intelligence' is concerned with systems that are modeled after social insect colonies, so that the different components of the system are queen, worker, soldier, etc. It can be viewed as ecological modeling in which the individual entities have extremely limited computing capacity and/or action sets, and in which there are very few types of entities. The premise of the field is that the rich behavior of social insect colonies arises not from the sophistication of any individual entity in the colony, but from the interaction among those entities. The objective of current research is to uncover kinds of interactions among the entity types that lead to pre-specified behavior of some sort.

More speculatively, the study of social insect colonies may also provide insight into how to achieve learning in large distributed systems. This is because at the level of the individual insect in a colony, very little (or no) learning takes place. However across evolutionary time-scales the social insect species as a whole functions as if the various individual types in a colony had "learned" their specific functions. The "learning" is the direct result of natural selection. (See the discussion on this topic in the subsection on ecological modeling.)

Swarm intelligences have been used to adaptively allocate tasks [33, 136], solve the traveling salesman problem [62, 63] and route data efficiently in dynamic networks [32, 201, 219] among others. However, there is no general framework for adapting swarm intelligences to maximize particular world utility functions. Accordingly, such intelligences generally need to be hand-tailored for each application.

1.2.5 Physics-Based Systems

Statistical Physics

Equilibrium statistical physics is concerned with the stable state character of large numbers of very simple physical objects, interacting according to well-specified local deterministic laws, with probabilistic noise processes superimposed [6, 188]. Typically there is no sense in which such systems can be said to have centralized control, since all particles contribute comparably to the overall dynamics.

Aside from mesoscopic statistical physics, the numbers of particles considered are usually huge (*e.g.*, 10^{23}), and the particles themselves are extraordinarily simple, typically having only a few degrees of freedom. Moreover, the noise processes usually considered are highly restricted, being

those that are formed by "baths", of heat, particles, and the like. Similarly, almost all of the field restricts itself to deterministic laws that are readily encapsulated in Hamilton's equations (Schrodinger's equation and its field-theoretic variants for quantum statistical physics). In fact, much of equilibrium statistical physics isn't even concerned with the dynamic laws by themselves (as for example is stochastic Markov processes). Rather it is concerned with invariants of those laws (*e.g.*, energy), invariants that relate the states of all of the particles. Deterministic laws without such readily-discoverable invariants are outside of the purview of much of statistical physics.

One potential use of statistical physics for collectives involves taking the systems that statistical physics analyzes, especially those analyzed in its condensed matter variant (*e.g.*, spin glasses [213, 214]), as simplified models of a class of collectives. This approach is used in some of the analyses of the El Farol Bar problem, also called the minority game (see below) [5, 48]. It is used more overtly in (for example) the work of Galam [90], in which the equilibrium coalitions of a set of "countries" are modeled in terms of spin glasses. This approach cannot provide a general collectives framework though. This is due to its not providing a general solution to arbitrary collectives inversion problems, being only concerned with the kinds of systems discussed above, and to its not employing RL algorithms.[7]

Another contribution that statistical physics can make is with the mathematical techniques it has developed for its own purposes, like mean field theory, self-averaging approximations, phase transitions, Monte Carlo techniques, the replica trick, and tools to analyze the thermodynamic limit in which the number of particles goes to infinity. Although such techniques have not yet been applied to collectives, they have been successfully applied to related fields. This is exemplified by the use of the replica trick to analyze two-player zero-sum games with random payoff matrices in the thermodynamic limit of the number of strategies in [27]. Other examples are the numeric investigation of iterated prisoner's dilemma played on a lattice [223], the analysis of stochastic games by expressing of deviation from rationality in the form of a "heat bath" [154], and the use of topological entropy to quantify the complexity of a voting system studied in [158].

Other quite recent work in the statistical physics literature is formally identical to that in other fields, but presents it from a novel perspective.

[7] In regard to the latter point however, it's interesting to speculate about recasting statistical physics as a collective, by viewing each of the particles in the physical system as running an "RL algorithm" that perfectly optimizes the "utility function" of its Lagrangian, given the "actions" of the other particles. In this perspective, many-particle physical systems are multi-stage games that are at Nash equilibrium in each stage. So for example, a frustrated spin glass is such a system at a Nash equilibrium that is not Pareto optimal.

A good example of this is [211], which is concerned with the problem of controlling a spatially extended system with a single controller, by using an algorithm that is identical to a simple-minded proportional RL algorithm (in essence, a rediscovery of RL).

Action Extremization

Much of the theory of physics can be cast as solving for the extremization of an actional, which is a functional of the worldline of an entire (potentially many-component) system across all time. The solution to that extremization problem constitutes the actual worldline followed by the system. In this way the calculus of variations can be used to solve for the worldline of a dynamic system. As an example, simple Newtonian dynamics can be cast as solving for the worldline of the system that extremizes a quantity called the 'Lagrangian', which is a function of that worldline and of certain parameters (*e.g.*, the 'potential energy') governing the system at hand. In this instance, the calculus of variations simply results in Newton's laws.

If we take the dynamic system to be a collective, we are assured that its worldline automatically optimizes a "global goal" consisting of the value of the associated actional. If we change physical aspects of the system that determine the functional form of the actional (*e.g.*, change the system's potential energy function), then we change the global goal, and we are assured that our collective optimizes that new global goal. Counter-intuitive physical systems, like the strings-and-springs systems that exhibit Braess' paradox [20], are simply systems for which the "world utility" implicit in our human intuition is extremized at a point different from the one that extremizes the system's actional.

The challenge in exploiting this to solve the design of collectives problem is in translating an arbitrary provided global goal for the collective into a parameterized actional. Note that that actional must govern the dynamics of the collective, and the parameters of the actional must be physical variables in the collective, variables whose values we can modify.

Active Walker Models

The field of active walker models [21, 100, 101] is concerned with modeling "walkers" (be they human walkers or instead simple physical objects) crossing fields along trajectories, where those trajectories are a function of several factors, including in particular the trails already worn into the field. Often the kind of trajectories considered are those that can be cast as solutions to actional extremization problems so that the walkers can be explicitly viewed as agents optimizing a private utility.

One of the primary concerns with the field of active walker models is how the trails worn in the field change with time to reach a final equilibrium state. The problem of how to design the cement pathways in the field (and other physical features of the field) so that the final paths actually

followed by the walkers will have certain desirable characteristics is then one of solving for parameters of the actional that will result in the desired worldline. This is a special instance of the inverse problem of how to design a collective.

Using active walker models this way to design collectives, like action extremization in general, probably has limited applicability. Also, it is not clear how robust such a design approach might be, or whether it would be scalable and exempt from the need for hand-tailoring.

1.2.6 Other Related Subjects

This subsection presents a "catch-all" of other fields that have little in common with one another and while either still nascent or not extremely closely related to collectives, bear some relation to collectives.

Stochastic Fields

An extremely well-researched body of work concerns the mathematical and numeric behavior of systems for which the probability distribution over possible future states conditioned on preceding states is explicitly provided. This work involves many aspects of Monte Carlo numerical algorithms [172], all of Markov Chains [80, 177, 215], and especially Markov fields, a topic that encompasses the Chapman-Kolmogorov equations [91] and its variants: Liouville's equation, the Fokker-Plank equation, and the Detailed-balance equation in particular. Non-linear dynamics is also related to this body of work (see the synopsis of iterated function systems below and the synopsis of cellular automata above), as is Markov competitive decision processes (see the synopsis of game theory above).

Formally, one can cast the problem of designing a collective as how to fix each of the conditional transition probability distributions of the individual elements of a stochastic field so that the aggregate behavior of the overall system is of a desired form.[8]

Amorphous computing and Control of Smart Matter

Amorphous computing grew out of the idea of replacing traditional computer design, with its requirements for high reliability of the components of

[8] In contrast, in the field of Markov decision processes, discussed in [45], the full system may be a Markov field, but the system designer only sets the conditional transition probability distribution of a few of the field elements at most, to the appropriate "decision rules". Unfortunately, it is hard to imagine how to use the results of this field to design collectives because of major scaling problems. Any decision process must accurately model likely future modifications to its own behavior — often an extremely daunting task [150]. What's worse, if multiple such decision processes are running concurrently in the system, each such process must also model the others, potentially needing to model them in their full complexity.

the computer, with a novel approach in which widespread unreliability of those components would not interfere with the computation [2, 1]. Some of its more speculative aspects are concerned with "how to program" a massively distributed, noisy system of components which may consist in part of biochemical and/or biomechanical components [131, 233]. Work here has tended to focus on schemes for how to robustly induce desired geometric dynamics across the physical body of the amorphous computer — issue that are closely related to morphogenesis, and thereby lend credence to the idea that biochemical components are a promising approach.

Especially in its limit of computers with very small constituent components, amorphous computing also is closely related to the fields of nanotechnology [64]. As the prospect of nanotechnology-driven mechanical systems gets more concrete, the daunting problem of how to robustly control, power, and sustain protean systems made up of extremely large sets of nano-scale devices looms more important [95, 96, 107]. If this problem were to be solved one would in essence have "smart matter". For example, one would be able to "paint" an airplane wing with such matter and have it improve drag and lift properties significantly.

Self Organizing Systems

The concept of self-organization and self-organized criticality [15] was originally developed to help understand why many distributed physical systems are attracted to critical states that possess long-range dynamic correlations in the large-scale characteristics of the system. It provides a powerful framework for analyzing both biological and economic systems. For example, natural selection (particularly punctuated equilibrium [68, 93]) can be likened to self-organizing dynamical system, and some have argued it shares many the properties (*e.g.*, scale invariance) of such systems [57]. Similarly, one can view the economic order that results from the actions of human agents as a case of self-organization [59]. The relationship between complexity and self-organization is a particularly important one, in that it provides the potential laws that allow order to arise from chaos [125].

Adaptive Control Theory

Adaptive control [7, 196], and in particular adaptive control involving locally weighted RL algorithms [9, 165], constitute a broadly applicable framework for controlling small, potentially inexactly modeled systems. Augmented by techniques in the control of chaotic systems [52, 60, 61], they constitute a very successful way of solving the "inverse problem" for such systems. Unfortunately, it is not clear how one could even attempt to scale such techniques up to the massively distributed systems of interest in collectives.

1.3 COIN Framework

The previous section provided a summary of different fields that address various issues pertinent to the field of collectives. In this section, we summarize the COIN (Collective Intelligence) framework, which is one of the first frameworks that aims to bridge the gap between the needs of the field of collectives and the extant analysis/design methods.[9]

1.3.1 Central Equation

Let Z be an arbitrary vector space whose elements z give the joint move of all agents in the system (i.e., z specifies the full "worldline" consisting of the actions/states of all the agents). The **world utility** $G(z)$, is a function of the full worldline, and we are concerned with the problem of finding the z that maximizes $G(z)$.

In addition to G, for each agent η, there is a **private utility function** $\{g_\eta\}$. The agents act to improve their individual private utility functions, even though, we, as system designers are only concerned with the value of the world utility G. To specify all agents other than η, we will use the notation $\hat{\eta}$.

Our uncertainty concerning the behavior of the system is reflected in a probability distribution over Z. Our ability to control the system consists of setting the value of some characteristic of the agents, e.g., setting the private functions of the agents. Indicating that value by s, our analysis revolves around the following central equation for $P(G \mid s)$, which follows from Bayes' theorem:

$$P(G \mid s) = \int d\vec{\epsilon}_G P(G \mid \vec{\epsilon}_G, s) \int d\vec{\epsilon}_g P(\vec{\epsilon}_G \mid \vec{\epsilon}_g, s) P(\vec{\epsilon}_g \mid s) \; , \qquad (1.1)$$

where $\vec{\epsilon}_g$ is the vector of the "intelligences" of the agents with respect to their associated private functions, and $\vec{\epsilon}_G$ is the vector of the intelligences of the agents with respect to G. Intuitively, these vectors indicate what percentage of η's actions would have resulted in lower utility.[10] In this chapter, we use intelligence vectors as decomposition variables for Equation 1.1.

Note that $\epsilon_{g_\eta}(z) = 1$ means that player η is fully rational at z, in that its move maximizes the value of its utility, given the moves of the players. In other words, a point z where $\epsilon_{g_\eta}(z) = 1$ for all players η is one that meets the definition of a game-theory Nash equilibrium. On the other hand, a z at which all components of $\vec{\epsilon}_G = 1$ is a local maximum of G (or more precisely, a critical point of the $G(z)$ surface). So if we can get these two vectors to be identical, then if the agents do well enough at maximizing

[9] The full COIN theory is presented in Chapter 2.

[10] Intelligence is formally defined in Chapter 2.

their private utilities we are assured we will be near a local maximum of G.

To formalize this, consider our decomposition of $P(G \mid s)$. If we can choose s so that the third conditional probability in the integrand, $P(\vec{\epsilon}_g \mid s)$, is peaked around vectors $\vec{\epsilon}_g$ all of whose components are close to 1 (that is agents are able to "learn" their tasks), then we have likely induced large private utility intelligences. If we can also have the second term, $P(\vec{\epsilon}_G \mid \vec{\epsilon}_g, s)$, be peaked about $\vec{\epsilon}_G$ equal to $\vec{\epsilon}_g$ (that is the private and world utilities are aligned), then $\vec{\epsilon}_G$ will also be large. Finally, if the first term in the integrand, $P(G \mid \vec{\epsilon}_G, s)$, is peaked about high G when $\vec{\epsilon}_G$ is large, then our choice of s will likely result in high G, as desired.

1.3.2 Factoredness and Learnability

For high values of G to be achieved, the private utility functions need to have two properties. [11] First, the private utility functions need to be "aligned with G", a need that is expressed in the second term of Equation 1.1. In particular, regardless of the details of the stochastic environment in which the agents operate, or of the details of the learning algorithms of the agents, if $\vec{\epsilon}_g$ equals $\vec{\epsilon}_G$ exactly *for all* z, the desired form for the second term in Equation 1.1 is assured. We call such a system **factored**. In game theory language, the private utility function Nash equilibria of a factored system are local maxima of G. In addition to this desirable equilibrium behavior, factored systems also automatically provide appropriate off-equilibrium incentives to the agents (an issue generally not considered in the game theory / mechanism design literature).

Second, we want the agents' private utility functions to have high **learnability**, intuitively meaning that an agent's utility should be sensitive to its own actions and insensitive to actions of others. This requirement that private utility functions have high "signal-to-noise" arises in the third term. As an example, consider a "team game" where the private utility functions are set to G. [56] Such a system is tautologically factored. However team games often have low learnability, because in a large system an agent will have a difficult time discerning the effects of its actions on G. As a consequence, each η may have difficulty achieving high g_η in such a system. Loosely speaking, agent η's learnability is the ratio of the sensitivity of g_η to η's actions to the sensitivity g_η to the actions of all other agents. So

[11] Non-game theory-based function maximization techniques like simulated annealing instead address how to have term 1 have the desired form. They do this by trying to ensure that the local maxima that the underlying system ultimately settles near have high G, by "trading off exploration and exploitation". One can combine such term-1-based techniques with the techniques presented here, The resultant hybrid algorithm, addressing all three terms, outperforms simulated annealing by over two orders of magnitude [240].

at a given state z, the higher the learnability, the more $g_\eta(z)$ depends on the move of agent η, i.e., the better the associated signal-to-noise ratio for η. Intuitively then, higher learnability means it is easier for η to achieve a large value of its utility.

1.3.3 Difference Utilities

It is possible to solve for the set of all private utilities that are factored with respect to a particular world utility. Unfortunately, in general it is not possible for a collective both to be factored and to have perfect learnability for all of its players (i.e., no dependence of any g_η on any agent other than η) for all of its agents [238]. However, consider **difference** utilities, which are of the form:

$$DU(z) = G(z) - \Gamma(f(z)) \ , \tag{1.2}$$

where $\Gamma(f)$ is independent of z_η. Such difference utilities are factored [238]. In addition, under usually benign approximations, learnability is maximized over the set of difference utilities by choosing

$$\Gamma(f(z)) = E(G \mid z_{\neg\eta}, s) \ , \tag{1.3}$$

up to an overall additive constant. We call the resultant difference utility the **Aristocrat** utility (AU). If each player η uses an appropriately rescaled version of the associated AU as its private utility function, then we have ensured good form for both terms 2 and 3 in Equation 1.1.

Using AU in practice is sometimes difficult, due to the need to evaluate the expectation value. Fortunately there are other utility functions that, while being easier to evaluate than AU, still are both factored and possess superior learnability to the team game utility, $g_\eta = G$. One such private utility function is the **Wonderful Life** Utility (WLU). The WLU for player η is parameterized by a pre-fixed **clamping parameter** CL_η chosen from among η's possible moves:

$$WLU_\eta \equiv G(z) - G(z_{\neg\eta}, CL_\eta) \ . \tag{1.4}$$

WLU is factored no matter what the choice of clamping parameter. Furthermore, while not matching the high learnability of AU, WLU usually has far better learnability than does a team game, because most of the "noise" due to other agents is removed from η's utility. Therefore, WLU generally results in better performance than does team game utilities [228, 238].

Figure 1.1 provides an example of clamping. As in that example, in many circumstances there is a particular choice of clamping parameter for agent η that is a "null" move for that agent, equivalent to removing that agent from the system. For such a clamping parameter WLU is closely related to the economics technique of "endogenizing a player's (agent's) externalities", for example with the Groves mechanism [174, 175, 87].

$$
\begin{array}{c} \\ \eta_1 \\ \eta_2 \\ \eta_3 \\ \eta_4 \end{array}
\begin{array}{c} z \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right] \end{array}
\quad \begin{array}{c} \\ \Longrightarrow \\ \text{Clamp} \\ \eta_2 \text{ to} \\ \text{“null”} \end{array} \quad
\begin{array}{c} (z_{\hat{\eta}_2}, \vec{0}) \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ \textit{0} & \textit{0} & \textit{0} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right] \end{array}
$$

$$
\begin{array}{c} \\ \Longrightarrow \\ \text{Clamp} \\ \eta_2 \text{ to} \\ \text{“average”} \end{array}
\begin{array}{c} (z_{\hat{\eta}_2}, \vec{a}) \\ \left[\begin{array}{ccc} 1 & 0 & 0 \\ \textit{.33} & \textit{.33} & \textit{.33} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{array}\right] \end{array}
$$

FIGURE 1.1. This example shows the impact of the clamping operation on the joint state of a four-agent system where each agent has three possible actions, and each such action is represented by a three-dimensional unary vector. The first matrix represents the joint state of the system z where agent 1 has selected action 1, agent 2 has selected action 3, agent 3 has selected action 1 and agent 4 has selected action 2. The second matrix displays the effect of clamping agent 2's action to the "null" vector (i.e., replacing z_{η_2} with $\vec{0}$). The third matrix shows the effect of instead clamping agent 2's move to the "average" action vector $\vec{a} = \{.33, .33, .33\}$, which amounts to replacing that agent's move with the "illegal" move of fractionally taking each possible move ($z_{\eta_2} = \vec{a}$).

However it is usually the case that using WLU with a clamping parameter that is as close as possible to the expected move defining AU results in far higher learnability than does clamping to the null move. Such a WLU is roughly akin to a mean-field approximation to AU.[12] For example, in Fig. 1.1, if the probabilities of player 2 making each of its possible moves was 1/3, then one would expect that a clamping parameter of $\vec{a}$ would be close to optimal. Accordingly, in practice use of such an alternative WLU derived as a "mean-field approximation" to AU almost always results in far better values of G than does the "endogenizing" WLU.

Intuitively, collectives having factored and highly learnable private utilities like AU and WLU can be viewed as akin to well-run human companies. G is the "bottom line" of the company, the players η are identified with the employees of that company, and the associated g_η given by the employees' performance-based compensation packages. For example, for a "factored company", each employee's compensation package contains incentives de-

[12] Formally, our approximation is exact only if the expected value of G equals G evaluated at the expected joint move (both expectations being conditioned on given moves by all players other than η). In general though, for relatively smooth G, we would expect such a mean-field approximation to AU, to give good results, even if the approximation does not hold exactly.

signed such that the better the bottom line of the corporation, the greater the employee's compensation. As an example, the CEO of a company wishing to have the private utilities of the employees be factored with G may give stock options to the employees. The net effect of this action is to ensure that what is good for the employee is also good for the company. In addition, if the compensation packages are "highly learnable", the employees will have a relatively easy time discerning the relationship between their behavior and their compensation. In such a case the employees will both have the incentive to help the company and be able to determine how best to do so. Note that in practice, providing stock options is usually more effective in small companies than in large ones. This makes perfect sense in terms of the formalism summarized above, since such options generally have higher learnability in small companies than they do in large companies, in which each employee has a hard time seeing how his/her moves affect the company's stock price.

1.3.4 Summary of COIN Results to Date

In earlier work, we tested the WLU for distributed control of network packet routing [241], achieving substantially better throughput than by using the best possible shortest-path-based system [241], even though that SPA-based system has information denied the agents in the WLU-based collective. In related work we have shown that use of the WLU automatically avoids the infamous Braess' paradox, in which adding new links can actually decrease throughput — a situation that readily ensnares SPA's [228, 239].

We have also applied the WLU to the problem of controlling communication across a constellation of satellites so as minimize the importance-weighted loss of scientific data flowing across that constellation [237]. We have also shown that agents using utility functions derived from the COIN framework significantly improve performance in the problem of job scheduling across a heterogeneous computing grid [227].

In addition we have explored COIN-based techniques on variants of congestion games [238, 242, 243], in particular of a more challenging variant of Arthur's El Farol bar attendance problem [5] (also known as the "minority game" [48]). In this work we showed that use of the WLU can result in performance *orders of magnitude* superior to that of team game utilities. We have also successfully applied COIN techniques to the problem of coordinating a set of autonomous rovers so as to maximize the importance-weighted value of a set of locations they visit [226].

Finally we have also explored applying COIN techniques to problems that are explicitly cast as search. These include setting the states of the spins in a spin glass to minimize energy; the conventional bin-packing problem of computer science, and a model of human agents connected in a small-world network who have to synchronize their purchase decisions [240].

1.4 Applications/Problems Driving Collectives

The previous sections focused on fields that provide solutions to problems arising in the field of collectives. To complement them, in this section we present three problems that are particularly suited to being approached from the field of collectives, and that provide fertile ground for testing novel theories of collectives.

1.4.1 El Farol Bar Problem (Minority Game)

The "El Farol" bar problem (also known as the minority game) and its variants provide a clean and simple testbed for investigating certain kinds of interactions among agents [5, 44, 47, 206]. In the original version of the problem, which arose in economics, at each time step (each "night"), each agent needs to decide whether to attend a particular bar. The goal of the agent in making this decision depends on the total attendance at the bar on that night. If the total attendance is below a preset capacity then the agent should have attended. Conversely, if the bar is overcrowded on the given night, then the agent should not attend. (Because of this structure, the bar problem with capacity set to 50% of the total number of agents is also known as the 'minority game'; each agent selects one of two groups at each time step, and those that are in the minority have made the right choice). The agents make their choices by predicting ahead of time whether the attendance on the current night will exceed the capacity and then taking the appropriate course of action.

What makes this problem particularly interesting is that it is impossible for each agent to be perfectly "rational", in the sense of correctly predicting the attendance on any given night. This is because if most agents predict that the attendance will be low (and therefore decide to attend), the attendance will actually high, while if they predict the attendance will be high (and therefore decide not to attend) the attendance will be low. (In the language of game theory, this essentially amounts to the property that there are no pure strategy Nash equilibria [49, 246].) Alternatively, viewing the overall system as a collective, it has a Prisoner's Dilemma-like nature, in that "rational" behavior by all the individual agents thwarts the global goal of maximizing total enjoyment (defined as the sum of all agents' enjoyment and maximized when the bar is exactly at capacity).

This frustration effect is a crisp example of the difficulty that can arise when agents try to model agents that are in their turn modeling the first agents. It is similar to what occurs in spin glasses in physics, and makes the bar problem closely related to the physics of emergent behavior in distributed systems [46, 47, 48, 248]. Researchers have also studied the dynamics of the bar problem to investigate economic properties like competition, cooperation and collective behavior and especially their relationship

to market efficiency [58, 122, 197].

1.4.2 Data Routing in a Network

Packet routing in a data network [28, 110, 212, 231, 127, 94] presents a particularly interesting domain for the investigation of collectives. In particular, with such routing:
(i) the problem is inherently distributed;
(ii) for all but the most trivial networks it is impossible to employ global control ;
(iii) the routers have only access to local information (routing tables);
(iv) it constitutes a relatively clean and easily modified experimental testbed; and
(v) there are potentially major bottlenecks induced by 'greedy' behavior on the part of the individual routers, which behavior constitutes a readily investigated instance of the Tragedy Of the Commons (TOC).

Many of the approaches to packet routing incorporate a variant on RL [39, 43, 51, 147, 152]. Q–routing is perhaps the best known such approach and is based on routers using reinforcement learning to select the best path [39]. Although generally successful, Q–routing is not a general scheme for inverting a global task. This is even true if one restricts attention to the problem of routing in data networks — there exists a global task in such problems, but that task is directly used to construct the algorithm.

A particular version of the general packet routing problem that is acquiring increased attention is the Quality of Service (QoS) problem, where different communication packets (voice, video, data) share the same bandwidth resource but have widely varying importances both to the user and (via revenue) to the bandwidth provider. Determining which packet has precedence over which other packets in such cases is not only based on priority in arrival time but more generally on the potential effects on the income of the bandwidth provider. In this context, RL algorithms have been used to determine routing policy, control call admission and maximize revenue by allocating the available bandwidth efficiently [43, 152].

Many researchers have exploited the noncooperative game theoretic understanding of the TOC in order to explain the bottleneck character of empirical data networks' behavior and suggest potential alternatives to current routing schemes [25, 67, 132, 133, 139, 141, 179, 180, 208]. Closely related is work on various "pricing"-based resource allocation strategies in congestable data networks [149]. This work is at least partially based upon current understanding of pricing in toll lanes, and traffic flow in general (see below). All of these approaches are particularly of interest when combined with the RL-based schemes mentioned just above. Due to these factors, much of the current research on a general framework for collectives is directed toward the packet-routing domain (see next section).

1.4.3 Traffic Theory

Traffic congestion typifies the Tragedy of the Commons public good problem: everyone wants to use the same resource, and all parties greedily trying to optimize their use of that resource not only worsens global behavior, but also worsens *their own* private utility (*e.g.*, if everyone disobeys traffic lights, everyone gets stuck in traffic jams). Indeed, in the well-known Braess' paradox [20, 54, 55, 134], keeping everything else constant — including the number and destinations of the drivers — but opening a new traffic path can *increase* everyone's time to get to their destination. (Viewing the overall system as an instance of the Prisoner's dilemma, this paradox in essence arises through the creation of a novel 'defect-defect' option for the overall system.) Greedy behavior on the part of individuals also results in very rich global dynamic patterns, such as stop and go waves and clusters [102, 103].

Much of traffic theory employs and investigates tools that have previously been applied in statistical physics [102, 129, 130, 183, 187] (see subsection above). In particular, the spontaneous formation of traffic jams provides a rich testbed for studying the emergence of complex activity from seemingly chaotic states [102, 104]. Furthermore, the dynamics of traffic flow is particular amenable to the application and testing of many novel numerical methods in a controlled environment [16, 29, 202]. Many experimental studies have confirmed the usefulness of applying insights gleaned from such work to real world traffic scenarios [102, 170, 169].

1.5 Challenge Ahead

Unfortunately, though they provide valuable insight on *some* aspects of collectives, none of the fields discussed above can be modified to encompass systems meeting all of the requirements of a "field" of collectives. This is not too surprising, since none of those fields were explicitly designed to design/analyze collectives, but rather touched on certain aspects of collectives.

What is needed is a fundamentally new look at this field, one that though may borrow from the various fields, will not simply extend an existing field that was not meant to analyze *general* collectives. There are many directions in which future work on collectives can and will proceed. It is a vast and rich area of research, and understanding the interaction among the various fields is essential in forging new directions.

1.6 References

[1] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight Jr., R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss. Amorphous computing. *Communications of the ACM*, 43(5), May 2000.

[2] H. Abelson and N. Forbes. Morphous-computing techniques may lead to intelligent materials. *Computers in Physics*, 12(6):520–522, 1998.

[3] M. R. Anderson and T. W. Sandholm. Leveled commitment contracts with myopic and strategic agents. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 36–45, 1998.

[4] K. Arrow and G. Debreu. The existence of an equilibrium for a competitive equilibrium. *Econometrica*, 22:265–290, 1954.

[5] W. B. Arthur. Complexity in economic theory: Inductive reasoning and bounded rationality. *The American Economic Review*, 84(2):406–411, May 1994.

[6] W. Ashcroft and N. D. Mermin. *Solid State Physics.* W. B. Saunders, Philadelphia, 1976.

[7] J.J. Astrom and B. Wittenmark. *Adaptive Control.* Addison–Wesley, 1994.

[8] C. G. Atkeson. Nonparametric model-based reinforcement learning. In *Advances in Neural Information Processing Systems - 10*, pages 1008–1014. MIT Press, 1998.

[9] C. G. Atkeson, S. A. Schaal, and A. W. Moore. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[10] R. J. Aumann. Correlated equilibrium as an expression of Bayesian rationality. *Econometrica*, 55(1):1–18, 1987.

[11] R.J. Aumann and S. Hart. *Handbook of Game Theory with Economic Applications, Volumes I and II.* North-Holland Press, 1992.

[12] R. Axelrod. *The Evolution of Cooperation.* Basic Books, NY, 1984.

[13] R. Axelrod. *The Complexity of Cooperation: Agent–Based Models of Competition and Collaboration.* Princeton University Press, NJ, 1997.

[14] P. Bak and K. Sneppen. Punctuated equilibrium and criticality in a simple model of evolution. *Physical Review Letters*, 71(24):4083–4086, 1993.

[15] P. Bak, C. Tang, and K. Wiesenfeld. Self-organized criticality. *Physical Review A*, 38:364, 1988.

[16] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, and Y. Sugiyama. Dynamical model of traffic congestion and numerical simulation. *Physical Review E*, 51(2):1035–1042, 1995.

[17] S. Bankes. Exploring the foundations of artificial societies: Experiments in evolving solutions to the iterated N–player prisoner's dilemma. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 337–342. MIT Press, 1994.

[18] Y. Bar-Yam, editor. *The Dynamics of Complex Systems*. Westview Press, 1997.

[19] T. Basar and G.J. Olsder. *Dynamic Noncooperative Game Theory*. Siam, Philadelphia, PA, 1999. Second Edition.

[20] T. Bass. Road to ruin. *Discover*, 13(5):56–61, May 1992.

[21] M. Batty. Predicting where we walk. *Nature*, 388:19–20, July 1997.

[22] E. Baum. Toward a model of mind as a laissez-faire economy of idiots. In L. Saitta, editor, *Proceedings of the 13th International Conference on Machine Learning*, pages 28–36. Morgan Kaufman, 1996.

[23] E. Baum. Toward a model of mind as an economy of agents. *Machine Learning*, 1999. (in Press).

[24] M. Begon, D.J. Thompshon, and M. Mortimer, editors. *Population Ecology : A Unified Study of Animals and Plants*. Blackwell Science Inc., 1996.

[25] A. M. Bell and W. A. Sethares. The El Farol problem and the internet: Congestion and coordination failure. In *Fifth International Conference of the Society for Computational Economics*, Boston, MA, 1999.

[26] J. Bendor and P. Swistak. The evolutionary advantage of conditional cooperation. *Complexity*, 4(2):15–18, 1996.

[27] J. Berg and A. Engel. Matrix games, mixed strategies, and statistical mechanics. *Physics Review Letters*, 81:4999–5002, 1998. preprint cond-mat/9809265.

[28] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[29] O Biham and A. A. Middleton. Self-organization and a dynamical transition in traffic-flow models. *Physical Review A*, 46(10):R6124–6127, 1992.

[30] K. Binmore. *Fun and Games: A Text on Game Theory*. D. C. Heath and Company, Lexington, MA, 1992.

[31] L. E. Blume and D. Easley. Optimality and natural selection in markets. pre-print: econ-wpa 9712003.pdf, 1997.

[32] E. Bonabeau, F. Henaux, S. Guerin, D. Snyders, P. Kuntz, and G. Theraulaz. Routing in telecommunications networks with "smart" and-like agents. (pre-print), 1999.

[33] E. Bonabeau, A. Sobkowski, G. Theraulaz, and J.-L. Deneubourg. Adaptive task allocation inspired by a model of division of labor of social insects. (pre-print), 1999.

[34] V. S. Borkar, S. Jain, and G. Rangarajan. Collective behaviour and diversity in economic communities: Some insights from an evolutionary game. In *Proceedings of the Workshop on Econophysics*, Budapest, Hungary, 1997.

[35] V. S. Borkar, S. Jain, and G. Rangarajan. Dynamics of individual specialization and global diversification in communities. *Complexity*, 3(3):50–56, 1998.

[36] C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge*, Holland, 1996.

[37] C. Boutilier. Learning conventions in multiagent stochastic domains using likelihood estimates. (pre-print), 1999.

[38] C. Boutilier, Y. Shoham, and M. P. Wellman. Editorial: Economic principles of multi-agent systems. *Artificial Intelligence Journal*, 94:1–6, 1997.

[39] J. A. Boyan and M. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in Neural Information Processing Systems - 6*, pages 671–678. Morgan Kaufman, 1994.

[40] J. M. Bradshaw, editor. *Software Agents*. MIT Press, 1997.

[41] R. A. Brooks. Intelligence without reason. In *Proceedings of the Twelveth International Joint Conference on Artificial Intelligence*, pages 569–595, 1991.

[42] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47:139–159, 1991.

[43] T. X. Brown, H. Tong, and S. Singh. Optimizing admission control while ensuring quality of service in multimedia networks via reinforcement learning. In *Advances in Neural Information Processing Systems - 11*. MIT Press, 1999.

[44] G. Caldarelli, M. Marsili, and Y. C. Zhang. A prototype model of stock exchange. *Europhysics Letters*, 40:479–484, 1997.

[45] A. R. Cassandra, L. P. Kaelbling, and M. L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1994.

[46] A. Cavagna. Irrelevance of memory in the minority game. preprint cond-mat/9812215, December 1998.

[47] D. Challet and Y. C. Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A*, 246(3-4):407, 1997.

[48] D. Challet and Y. C. Zhang. On the minority game: Analytical and numerical studies. *Physica A*, 256:514, 1998.

[49] J. Cheng. The mixed strategy equilibria and adaptive dynamics in the bar problem. Technical report, Santa Fe Institute Computational Economics Workshop, 1997.

[50] D. R. Cheriton and K. Harty. A market approach to operating system memory allocation. In S.E. Clearwater, editor, *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific, 1995.

[51] S. P. M. Choi and D. Y. Yeung. Predictive Q-routing: A memory based reinforcement learning approach to adaptive traffic control. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 945–951. MIT Press, 1996.

[52] D. J. Christini and J. J. Collins. Using noise and chaos control to control nonchaotic systems. *Physical Review E*, 52(6):5806–5809, 1995.

[53] C. Claus and C. Boutilier. The dynamics of reinforcement learning cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746–752, Madison, WI, June 1998.

[54] J. E. Cohen and C. Jeffries. Congestion resulting from increased capacity in single-server queueing networks. *IEEE/ACM Transactions on Networking*, 5(2):305–310, 1997.

[55] J. E. Cohen and F. P. Kelly. A paradox of congestion in a queuing network. *Journal of Applied Probability*, 27:730–734, 1990.

[56] R. H. Crites and A. G. Barto. Improving elevator performance using reinforcement learning. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 1017–1023. MIT Press, 1996.

[57] J. de Boer, B. Derrida, H. Flyvberg, A. D. Jackson, and T. Wettig. Simple model of self-organized biological evolution. *Physical Review Letters*, 73(6):906–909, 1994.

[58] M. A. R. de Cara, O. Pla, and F. Guinea. Competition, efficiency and collective behavior in the "El Farol" bar model. *European Physical Journal B*, 10:187, 1999.

[59] A. de Vany. The emergence and evolution of self-organized coalitions. In M. Gilli, editor, *Computational Methods in Economics*. Kluwer Scientific Publishers, 1999. (to appear).

[60] W. L. Ditto, S. N. Rauseo, and M. L. Spano. Experimental control of chaos. *Phys. Rev. Let.*, 65:3211, 1990.

[61] W. L. Ditto and K. Showalter. Introduction: Control and synchronization of chaos. *Chaos*, 7(4):509–511, 1997.

[62] M. Dorigo and L. M. Gambardella. Ant colonies for the travelling salesman problem. *Biosystems*, 39, 1997.

[63] M. Dorigo and L. M. Gambardella. Ant colony systems: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[64] K. E. Drexler. *Nanosystems: Molecular Machinery, Manufacturing, and Computation*. John Wiley and Sons, 1992.

[65] B. Drossel. A simple model for the formation of a complex organism. preprint adap-org/9811002, November 1998.

[66] J. Eatwell, Milgate M., and Newman P. *The New Palgrave Game Theory*. Macmillan Press, 1989.

[67] A. A. Economides and J. A. Silvester. Multi-objective routing in integrated services networks: A game theory approach. In *IEEE Infocom '91: Proceedings of the Conference on Computer Communication*, volume 3, 1991.

[68] N. Eldredge and S. J. Gould. Punctuated equilibria: An alternative to phyletic gradualism. In J. M. Schopf, editor, *Models in Paleobiology*, pages 82–115. Greeman, Cooper, 1972.

[69] C. M. Ellison. The Utah TENEX scheduler. *Proceedings of the IEEE*, 63:940–945, 1975.

[70] J. M. Epstein. Zones of cooperation in demographic prisoner's dilemma. *Complexity*, 4(2):36–48, 1996.

[71] J. M. Epstein. *Nonlinear Dynamics, Mathematical Biology, and Social Science.* Addison Wesley, Reading, MA, 1997.

[72] J. M. Epstein and R. Axtell. *Growing Artificial Societies: Social Sciences from the Bottom Up.* MIT Press, Cambridge, MA, 1996.

[73] N. Feltovich. Equilibrium and reinforcement learning with private information: An experimental study. pre-print, Dept. of Economics, U. of Houston, July 1997.

[74] J. Ferber. Reactive distributed artificial intelligence: Principles and applications. In G. O-Hare and N. Jennings, editors, *Foundations of Distributed Artificial Intelligence*, pages 287–314. John Wiley and Sons, 1996.

[75] D. F. Ferguson, C. Nikolaou, and Y. Yemini. An economy for flow control in computer networks. In *IEEE Infocom '89*, pages 110–118, 1989.

[76] S. G. Ficici and J. B. Pollack. Challenges in coevolutionary learning: Arms-race dynamics, open-endedness, and mediocre stable states. In C. Adami et al., editor, *Artificial Life VI*, pages 238–247. MIT Press, 1998.

[77] J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer–Verlag, 1997.

[78] D. B. Fogel. An overview of evolutionary programming. In L. D. Davis, K. De Jong, M. D. Vose, and L. D. Whitley, editors, *Evolutionary Algorithms*, pages 89–109. Springer, 1997.

[79] C. L. Forgy. RETE: A fast algorithm for the many pattern/many object patent match problem. *Artificial Intelligence*, 19(1):17–37, 1982.

[80] D. Freedman. *Markov Chains.* Springer–Verlag, 1983.

[81] E. Friedman. Strategic properties of heterogeneous serial cost sharing. In *Mathematical Social Sciences.* 2000.

[82] E. Friedman and D. C. Parkes. Pricing WiFi at Starbucks – Issues in online mechanism design. In *Fourth ACM Conf. on Electronic Commerce*, 2003.

[83] E. Friedman and S. Shenker. Learning and implementation in the Internet. available from (www.orie.cornell.edu/∼friedman), 2002.

[84] J. W. Friedman. *Game Theory with Applications to Economics.* Oxford University Press, New York, NY, 1986.

[85] D. Fudenberg and D. K. Levine. Steady state learning and Nash equilibrium. *Econometrica*, 61(3):547–573, 1993.

[86] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games.* MIT Press, Cambridge, MA, 1998.

[87] D. Fudenberg and J. Tirole. *Game Theory.* MIT Press, Cambridge, MA, 1991.

[88] Gabora. Autocatalytic closure in a cognitive system: A tentative scenario for the origin of culture. *Psycoloquy*, 9(67), December 1998.

[89] V. V Gafiychuk. Distributed self–regulation induced by negative feedbacks in ecological and economic systems. pre-print, adap-org/98110011, November 1998.

[90] S. Galam. Spontaneous coalition forming: A model from spin glass. pre-print cond-mat/9901022, January 1999.

[91] C.W. Gardiner. *Handbook of Stochastic Methods.* Springer–Verlag, New York, NY, 1985.

[92] C. V. Goldman and J. S. Rosenschein. Emergent coordination through the use of cooperative state–changing rules. (pre-print), 1999.

[93] S. J. Gould and N. Eldredge. Punctuated equilibria: the tempo and mode of evolution reconsidered. *Paleobiology*, 3:115–151, 1977.

[94] W. Grover. Self organizing broad band transport networks. *Proceedings of the IEEE*, 85(10):1582–1611, 1997.

[95] O. Guenther, T. Hogg, and B. A. Huberman. Learning in multiagent control of smart matter. In *AAAI-97 Workshop on Multiagent Learning*, 1997.

[96] O. Guenther, T. Hogg, and B. A. Huberman. Market organizations for controlling smart matter. In *Proceedings of the International Conference on Computer Simulation and Social Sciences*, 1997.

[97] E. A. Hansen, A. G. Barto, and S. Zilberstein. Reinforcement learning for mixed open-loop and closed loop control. In *Advances in Neural Information Processing Systems - 9*, pages 1026–1032. MIT Press, 1998.

[98] I Hanski. Be diverse, be predictable. *Nature*, 390:440–441, 1997.

[99] A. Hastings. *Population Biology : Concepts and Models.* Springer–Verlag, 1997.

[100] D. Helbing, J. Keltsch, and P. Molnar. Modeling the evolution of the human trail systems. *Nature*, 388:47–49, July 1997.

[101] D. Helbing, F. Schweitzer, J. Keltsch, and P. Molnar. Active walker model for the formation of human and animal trail systems. *Physical Review E*, 56(3):2527–2539, 1997.

[102] D. Helbing and M. Treiber. Jams, waves, and clusters. *Science*, 282:200–201, December 1998.

[103] D. Helbing and M. Treiber. Phase diagram of traffic states in the presence of inhomogeneities. *Physics Review Letters*, 81:3042, 1998.

[104] M. Herrmann and B. S. kerner. Local cluster effect in different traffic flow models. *Physica A*, 225:163–168, 1998.

[105] Yu-Chi Ho. Team decision theory and information structures. *Proceedings of the IEEE*, 68(644-654), 1980.

[106] T. Hogg and B. A. Huberman. Achieving global stability through local controls. In *Proceedings of the Sixth IEEE Symposium on Intelligent Control*, pages 67–72, 1991.

[107] T. Hogg and B. A. Huberman. Controlling smart matter. *Smart Materials and Structures*, 7:R1–R14, 1998.

[108] J. Holland and J. H. Miller. Artificial adaptive agents in economic theory. *American Economic Review*, 81:365–370, May 1991.

[109] J. H. Holland, editor. *Adaptation in Natural and Artificial Systems.* MIT Press, 1993.

[110] M.-T. T. Hsiao and A. A. Lazar. Optimal flow control of multi-class queueing networks with decentralized information. In *IEEE Infocom '89*, pages 652–661, 1987.

[111] J. Hu and M. P. Wellman. Self-fulfilling bias in multiagent learning. In *Proceedings of the Second International Conference on Multiagent Systems*, pages 118–125, 1996.

[112] J. Hu and M. P. Wellman. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 242–250, June 1998.

[113] J. Hu and M. P. Wellman. Online learning about other agents in a dynamic multiagent system. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 239–246, May 1998.

[114] M. Huber and R. A. Grupen. Learning to coordinate controllers – reinforcement learning on a control basis. In *Proceedings of the 15th International Conference of Artificial Intelligence*, volume 2, pages 1366–1371, 1997.

[115] B. A. Huberman, editor. *The Ecology of Computation.* North-Holland, Amsterdam, 1988.

[116] B. A. Huberman and S. H. Clearwater. A multi-agent system for controlling building environments. In *Proceedings of the International Conference on Multiagent Systems*, pages 171–176, 1995.

[117] B. A. Huberman and T. Hogg. The behavior of computational ecologies. In *The Ecology of Computation*, pages 77–115. North-Holland, 1988.

[118] M. E. Huhns, editor. *Distributed Artificial Intelligence.* Pittman, London, 1987.

[119] R. V. Iyer and S. Ghosh. DARYN, a distributed decision–making algorithm for railway networks: Modeling and simulation. *IEEE transaction of Vehicular Technology*, 44(1):180–191, 1995.

[120] P. Jefferies, M. L. Hart, and N. F. Johnson. Deterministic dynamics in the minority game. *Physical Review E*, 65 (016105), 2002.

[121] N. R. Jennings, K. Sycara, and M. Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1:7–38, 1998.

[122] N. F. Johnson, S. Jarvis, R. Jonson, P. Cheung, Y. R. Kwong, and P. M. Hui. Volatility and agent adaptability in a self-organizing market. preprint cond-mat/9802177, February 1998.

[123] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996.

[124] E. Kalai and E. Lehrer. Rational learning leads to Nash equilibrium. *Econometrica*, 61(5):1019–1045, 1993.

[125] S. A. Kauffman. *At Home in the Universe: The Search for the Laws of Self-Organization and Complexity.* Oxford University Press, 1995.

[126] L. Keller and H. K. Reeve. Familiarity breeds cooperation. *Nature*, 394:121–122, 1998.

[127] F. P. Kelly. Modeling communication networks, present and future. *Philosophical Trends Royal Society of London A*, 354:437–463, 1996.

[128] J. O. Kephart, J. E. Hanson, and J. Sairamesh. Price and niche wars in a free-market economy of software agents. *Artificial Life*, 4:1–13, 1998.

[129] B. S. Kerner, P. Konhauser, and M. Schilke. Deterministic spontaneous appearance of traffic jams in slightly inhomogeneous traffic flow. *Physical Review E*, 51(6):6243–6246, 1995.

[130] B. S. Kerner and H. Rehborn. Experimental properties of complexity in traffic flow. *Physical Review E*, 53(5):R4275–4278, 1996.

[131] T. F. Knight and G. J. Sussman. Cellular gate technology. In *Proceedings of the First International Conference on Unconventional Models of Computation*, Auckland, NZ, January 1998.

[132] Y. A. Korilis, A. A. Lazar, and A. Orda. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking*, 5(1):161–173, 1997.

[133] Y. A. Korilis, A. A. Lazar, and A. Orda. Capacity allocation under noncooperative routing. *IEEE Transactions on Automatic Control*, 42(3):309–325, 1997.

[134] Y. A. Korilis, A. A. Lazar, and A. Orda. Avoiding the Braess paradox in noncooperative networks. *Journal of Applied Probability*, 36:211–222, 1999.

[135] S. Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, pages 79–97, 1997.

[136] M.J.B. Krieger, J.-B. Billeter, and L. Keller. Ant-like task allocation and recruitment in cooperative robots. *Nature*, 406:992–995, 2000.

[137] V. Krishna and P. Motty. Efficient mechanism design. (pre-print), 1997.

[138] J. F. Kurose and R. Simha. A microeconomic approach to optimail resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 35(5):705–717, 1989.

[139] R. J. La and V. Anantharam. Optimal routing control: Game theoretic approach. (submitted to IEEE transactions on Automatic Control), 1999.

[140] M. Lauer and M. Riedmiller. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Seventeenth International Machine Learning Conference*, pages 535–542. Morgan Kauffman, 2000.

[141] A. A. Lazar, A. Orda, and D. E. Pendarakis. Capacity allocation under noncooperative routing. *IEEE Transactions on Networking*, 5(6):861–871, 1997.

[142] A. A. Lazar and N. Semret. Design, analysis and simulation of the progressive second price auction for network bandwidth sharing. Technical Report 487-98-21 (Rev 2.10), Columbia University, April 1998.

[143] T. S. Lee, S. Ghosh, J. Liu, X. Ge, and A. Nerode. A mathematical framework for asynchronous, distributed, decision–making systems with semi–autonomous entities: Algorithm synthesis, simulation, and evaluation. In *Fourth International Symposium on Autonomous Decentralized Systems*, Tokyo, Japan, 1999.

[144] T. M. Lenton. Gaia and natural selection. *Nature*, 394:439–447, 1998.

[145] K. Lerman and A. Galstyan. Mathematical model of foraging in a group of robots: Effect of interference. *Autonomous Robots*, 13(2), 2002.

[146] M. L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 157–163, 1994.

[147] M. L. Littman and J. Boyan. A distributed reinforcement learning scheme for network routing. In *Proceedings of the 1993 International Workshop on Applications of Neural Networks to Telecommunications*, pages 45–51, 1993.

[148] R. D. Luce and H. Raiffa. *Games and Decisions.* Dover Press, 1985.

[149] J. K. MacKie-Mason and R. V. Hal. Pricing congestible network resources. *IEEE Journal on Selected Areas of Communications*, 13(7):1141–1149, 1995.

[150] W. G. Macready and D. H. Wolpert. Bandit problems and the exploration/exploitation tradeoff. *IEEE Transactions on Evolutionary Computation*, 2:2–22, 1998.

[151] P. Maes. *Designing Autonomous Agents.* MIT Press, Cambridge, MA, 1990.

[152] P. Marbach, O. Mihatsch, M. Schulte, and J. Tsisiklis. Reinforcement learning for call admission control and routing in integrated service networks. In *Advances in Neural Information Processing Systems - 10*, pages 922–928. MIT Press, 1998.

[153] J. Marschak and R. Radner. *Economic Theory of Teams*. Yale University Press, New Haven, CO, 1972.

[154] M. Marsili and Y.-C. Zhang. Stochastic dynamics in game theory. preprint cond-mat/9801309, January 1998.

[155] J. Maynard Smith. *Evolution and the Theory of Games*. Cambridge University Press, 1982.

[156] D. McFarland. Toward robot cooperation. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 440–443. MIT Press, 1994.

[157] M. Mesterton-Gibbons, J.H. Marden, and L.A. Dugatkin. On wars of attrition without assessment. *Journal of Theoretical Biology*, 181:65–83, 1992.

[158] D. A. Meyer and T. A. Brown. Statistical mechanics of voting. *Physics Review Letters*, 81(8):1718–1721, 1998.

[159] J. H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87–112, 1996.

[160] J. H. Miller. Evolving information processing organizations. (preprint), 1996.

[161] J. H. Miller and J. Andreoni. Auctions with adaptive artificial agents. *Journal of Games and Economic Behavior*, 10:39–64, 1995.

[162] J. H. Miller, C. Butts, and D. Rode. Communication and cooperation. (pre-print), 1998.

[163] M. Minsky. *The Society of Mind*. Simon and Schuster, 1988.

[164] J. Mirrlees. An exploration in the theory of optimal income taxation. *Review of Economic Studies*, 38:175–208, 1974.

[165] A. W. Moore, C. G. Atkeson, and S. Schaal. Locally weighted learning for control. *Artificial Intelligence Review*, 11:75–113, 1997.

[166] R. Munos and P. Bourgine. Reinforcement learning for continuous stochastic control problems. In *Advances in Neural Information Processing Systems - 10*, pages 1029–1035. MIT Press, 1998.

[167] R. Nagpal. Programmable pattern-formation and scale-independence. In *Proceedings of the 4th International Conference on Complex Systems*, New Hampshire, June 2002.

[168] R. Nagpal. Programmable self-assembly using biologically-inspired multiagent control. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, July 2002.

[169] K. Naigel. Experiences with iterated traffic microsimulations in dallas. pre-print adap-org/9712001, December 1997.

[170] K. Naigel, P. Stretz, M. Pieck, S. Leckey, R. Donnelly, and C. Barrett. TRANSIMS traffic flow characteristics. pre-print adap-org/9710003, October 1997.

[171] J. F. Nash. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(48-49), 1950.

[172] R. M. Neal. *Bayesian Learning for Neural Networks, Lecture Notes in Statistics, No. 118.* Springer–Verlag, New York, 1996.

[173] A. Neyman. Bounded complexity justifies cooperation in the finitely repeated prisoner's dilemma. *Economics Letters*, 19:227–230, 1985.

[174] W. Nicholson. *Microeconomic Theory.* The Dryden Press, seventh edition, 1998.

[175] N. Nisan and A. Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35:166–196, 2001.

[176] S. I. Nishimura and T. Ikegami. Emergence of collective strategies in a prey-predator game model. *Artificial Life*, 3:243–360, 1997.

[177] J. Norris. *Markov Chains.* Cambridge University Press, 1998.

[178] M. A. Nowak and K. Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393:573–577, 1998.

[179] S. Olafsson. Games on networks. *Proceedings of the IEEE*, 85(10):1556–1562, 1997.

[180] A. Orda, R. Rom, and M. Sidi. Minimum delay routing in stochastic networks. *IEEE/ACM Transactions on Networking*, 1(2):187–198, 1993.

[181] D. C. Parkes. *Iterative Combinatorial Auctions: Theory and Practice.* PhD thesis, University of Pennsylvania, 2001.

[182] D. C. Parkes. Price-based information certificates for minimal-revelation combinatorial auctions. In *Agent Mediated Electronic Commerce IV: Designing Mechanisms and Systems*, volume 2531 of *Lecture Notes in Artificial Intelligence*. 2002.

[183] L. A. Pipes. An operational analysis of traffic dynamics. *Journal of Applied Physics*, 24(3):274–281, 1953.

[184] G. A. Polls. Stability is woven by complex webs. *Nature*, 395:744–745, 1998.

[185] M. Potters, R. Cont, and J.-P. Bouchaud. Financial markets as adaptive ecosystems. preprint cond-mat/9609172 v2, June 1997.

[186] D. Prokhorov and D. Wunsch. Adaptive critic design. *IEEE Transactions on Neural Networks*, 8(5):997–1007, 1997.

[187] Z. Qu, F. Xie, and G. Hu. Spatiotemporal on-off intermittency by random driving. *Physical Review E*, 53(2):R1301–1304, 1996.

[188] F. Reif. *Fundamentals of Statistical and Thermal Physics*. McGraw–Hill, 1965.

[189] E. Rich and K. Knight. *Artificial Intelligence*. McGraw-Hill, Inc., second edition, 1991.

[190] A.E. Roth and I. Erev. Learning in extensive-form games: experimental data and simple dynamic models in the intermediate term. *Games and economic behavior*, 8:164–212, 1995.

[191] A. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Reseach and Development*, 3:210–229, 1959.

[192] T. Sandholm and R. Crites. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems*, 37:147–166, 1995.

[193] T. Sandholm, K. Larson, M. Anderson, O. Shehory, and F. Tohme. Anytime coalition structure generation with worst case guarantees. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 46–53, 1998.

[194] T. Sandholm and V. R. Lesser. Issues in automated negotiations and electronic commerce: extending the contract net protocol. In *Proceedings of the Second International Conference on Multi-Agent Systems*, pages 328–335. AAAI Press, 1995.

[195] T. Sandholm and V. R. Lesser. Coalitions among computationally bounded agents. *Artificial Intelligence*, 94:99–137, 1997.

[196] S. Sastry and M. Bodson. *Adaptive Control, Stability, Convergence, and Robustness*. Prentice Hall, 1989.

[197] R. Savit, R. Manuca, and R. Riolo. Adaptive competition, market efficiency, phase transitions and spin-glasses. preprint cond-mat/9712006, December 1997.

[198] A. Schaerf, Y. Shoham, and M. Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 162:475–500, 1995.

[199] J. Schmidhuber, J. Zhao, and N. N. Schraudoiph. Reinforcement learning with self-modifying policies. In S. Thrun and L. Pratt, editors, *Learning to Learn*, pages 293–309. Kluwer, 1997.

[200] J. Schmidhuber, J. Zhao, and M. Wiering. Shifting inductive bias with success-story algorithm, adaptive Levin search, and incremental self-improvement. *Machine Learning*, 28:105–130, 1997.

[201] R. Schoonderwoerd, O. Holland, and J. Bruten. Ant-like agents for load balancing in telecommunication networks. In *Autonomous Agents 97*, pages 209–216. MIT Press, 1997.

[202] M. Schreckenberg, A. Schadschneider, K. Nagel, and N. Ito. Discrete stochastic models for traffic flow. *Physical Review E*, 51(4):2939–2949, 1995.

[203] J. Schull. Are species intelligent? *Behavioral and Brain Sciences*, 13:63–108, 1990.

[204] S. Sen. *Multi-Agent Learning: Papers from the 1997 AAAI Workshop (Technical Report WS-97-03.* AAAI Press, Menlo Park, CA, 1997.

[205] S. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. (pre-print), 1999.

[206] W. A. Sethares and A. M. Bell. An adaptive solution to the El Farol problem. In *Proceedings. of the Thirty-Sixth Annual Allerton Conference on Communication, Control, and Computing*, Allerton, IL, 1998. (Invited).

[207] R. Sethi. Stability of equilibria in games with procedural rational players. pre-print, Dept of Economics, Columbia University, November 1998.

[208] S. J. Shenker. Making greed work in networks: A game-theoretic analysis of switch service disciplines. *IEEE Transactions on Networking*, 3(6):819–831, 1995.

[209] Y. Shoham and K. Tanaka. A dynamic theory of incentives in multi-agent systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1997.

[210] J. Sidel, P. M. Aoki, S. Barr, A. Sah, C. Staelin, M. Stonebreaker, and Yu A. Data replication in mariposa. In *Proceedings of the 12th International Conference on Data Engineering*, 1996.

[211] S. Sinha and N. Gupte. Adaptive control of spatially extended systems: targeting spatiotemporal patterns and chaos. *Physical Review E*, 58(5):R5221–5224, 1998.

[212] W. Stallings. *Data and Computer Communications.* MacMillian Publishing Co., New York, 1994.

[213] J. Stein. Critical exponents of the $u(n)$ vector spin glasses. *Europhysics Letters*, 34(9):717–721, 1996.

[214] J. Stein. Critical properties of a spin glass with anisotropic dzyaloshinskii-moriya interaction. *J. Phys. A*, 29:963–971, 1996.

[215] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, 1995.

[216] P. Stone. TPOT-RL applied to network routing. In *Proceedings of the Seventeenth International Machine Learning Conference*, pages 935–942. Morgan Kauffman, 2000.

[217] P. Stone and M. Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8(3), 2000.

[218] M. Stonebreaker, P. M. Aoki, R. Devine, W. Litwin, and M. Olson. Mariposa: A new architecture for distributed data. In *Proceedings of the 10th International Conference on Data Engineering*, 1994.

[219] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence*, pages 832–838, 1997.

[220] R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

[221] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[222] K. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

[223] G. Szabo and C. Toke. Evolutionary prisoner's dilemma game on a square lattice. *Physical Review E*, 58(1):69–73, 1998.

[224] G. Tesauro. Practical issues in temporal difference learning. *Machine Learning*, 8:33–53, 1992.

[225] P. Tucker and F. Berman. On market mechanisms as a software techniques. Technical Report CS96–513, University of California, San Diego, December 1996.

[226] K. Tumer, A. Agogino, and D. Wolpert. Learning sequences of actions in collectives of autonomous agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 378–385, Bologna, Italy, July 2002.

[227] K. Tumer and J. Lawson. Collectives for multiple resource job scheduling across heterogeneous servers. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Melbourne, Australia, July 2003.

[228] K. Tumer and D. H. Wolpert. Collective intelligence and Braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 104–109, Austin, TX, 2000.

[229] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[230] C. A. Waldspurger, T. Hogg, B. A. Huberman, J. O. Kephart, and W. S. Stornetta. Spawn: A distributed computational economy. *IEEE transactions of Software engineering*, 18(2):103–117, 1992.

[231] J. Walrand and P. Varaiya. *High–Performance Communication Networks*. Morgan Kaufmann, San Fransisco, CA, 1996.

[232] C. Watkins and P. Dayan. Q-learning. *Machine Learning*, 8(3/4):279–292, 1992.

[233] R. Weiss, G. Homsy, and R. Nagpal. Programming biological cells. In *Proceedings of the 8th International Conference on Architectural Support for Programming Languages and Operating Systems*, San Jose, NZ, 1998.

[234] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. In *Journal of Artificial Intelligence Research*, 1993.

[235] M. P. Wellman. A computational market model for distributed configuration design. In *Proceedings of the 12th National Conference on Artificial Intelligence*, 1994.

[236] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. Best Paper Award.

[237] D. H. Wolpert, J. Sill, and K. Tumer. Reinforcement learning in distributed domains: Beyond team games. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 819–824, Seattle, WA, 2001.

[238] D. H. Wolpert and K. Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(2/3):265–279, 2001.

[239] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002.

[240] D. H. Wolpert, K. Tumer, and E. Bandari. Improving search algorithms by using intelligent coordinates. 2003. submitted.

[241] D. H. Wolpert, K. Tumer, and J. Frank. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, pages 952–958. MIT Press, 1999.

[242] D. H. Wolpert, K. Wheeler, and K. Tumer. General principles of learning-based multi-agent systems. In *Proceedings of the Third International Conference of Autonomous Agents*, pages 77–83, 1999.

[243] D. H. Wolpert, K. Wheeler, and K. Tumer. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters*, 49(6), March 2000.

[244] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. *Proceedings of the XI International Congress of Genetics*, 8:209–222, 1932.

[245] H. P. Young. The evolution of conventions. *Econometrica*, 61(1):57–84, 1993.

[246] E. Zambrano. Rationalizable bounded rational behavior. (pre-print), 1999.

[247] W. Zhang and T. G. Dietterich. Solving combinatorial optimization tasks by reinforcement learning: A general methodology applied to resource-constrained scheduling. *Journal of Artificial Intelligence Research*, 2000.

[248] Y. C. Zhang. Modeling market mechanism with evolutionary games. *Europhysics Letters*, March/April 1998.

[249] G. Zlotkin and J. S. Rosenschein. Coalition, cryptography, and stability: Mechanisms for coalition formation in task oriented domains. (pre-print), 1999.

13

Dynamics of Large Autonomous Computational Systems

Tad Hogg and Bernardo A. Huberman

ABSTRACT Distributed large scale computation gives rise to a wide range of behaviors, from the simple to the chaotic. This diversity of behaviors stems from the fact that the agents and programs have incomplete knowledge and imperfect information on the state of the system. We describe an instantiation of such systems based on market mechanisms which provides an interesting example of autonomous control. We also show that when agents choose among several resources, the dynamics of the system can be oscillatory and even chaotic. Furthermore, we describe a mechanism for achieving global stability through local controls.

13.1 Introduction

It is by now an established fact that computation has become widely distributed over the world, both as networked computers and embedded systems, such as the processing power available in automobiles, mobile phones and robots. And in distributed form the capabilities of the system as a whole are much greater than those of single components. This is because of the ability of a distributed system to share information, resources and to parcel the computation in efficient ways.

The effective use of distributed computation is a challenging task, since the processes must obtain resources in a dynamically changing environment and be designed to collaborate despite a variety of asynchronous and unpredictable changes. For instance, the lack of global perspectives for determining resource allocation requires a very different approach to system-level programming and the creation of suitable languages. Even implementing reliable methods whereby processes can compute in machines with diverse characteristics is difficult.

As these distributed systems grow, they become a community of concurrent processes, or a *computational ecosystem* [9], which, in their interactions, strategies, and lack of perfect knowledge, are analogous to biological ecosystems and human economies. Since all of these systems consist of

a large number of independent actors competing for resources, this analogy can suggest new ways to design and understand the behavior of these emerging computational systems. In particular, these existing systems have methods to deal successfully with coordinating asynchronous operations in the face of imperfect knowledge. These methods allow the system as a whole to adapt to changes in the environment or disturbances to individual members, in marked contrast to the brittle nature of most current computer programs which often fail completely if there is even a small change in their inputs or error in the program itself. To improve the reliability and usefulness of distributed computation, it is therefore of interest to examine the extent to which this analogy can be exploited.

Statistical mechanics, based on the law of large numbers, has taught us that many universal and generic features of large systems can be quantitatively understood as approximations to the average behavior of infinite systems. Although such infinite models can be difficult to solve in detail, their overall qualitative features can be determined with a surprising degree of accuracy. Since these features are universal in character and depend only on a few general properties of the system, they can be expected to apply to a wide range of actual configurations. This is the case when the number of relevant degrees of freedom in the system, as well as the number of interesting parameters, is small. In this situation, it becomes useful to treat the unspecified internal degrees of freedom as if they are given by a probability distribution. This implies assuming a lack of correlations between the unspecified and specified degrees of freedom. This assumption has been extremely successful in statistical mechanics. It implies that although degrees of freedom may change according to purely deterministic algorithms, the fact that they are unspecified makes them appear to an outside observer as effectively random.

Consider, for instance, massively parallel systems which are desired to be robust and adaptable. They should work in the presence of unexpected errors and with changes in the environment in which they are embedded (i.e., fail soft). This implies that many of the system's internal degrees of freedom will be allowed to adjust by taking on a range of possible configurations. Furthermore, their large size will necessarily enforce a perspective which concentrates on a few relevant variables. Although these considerations suggest that the assumptions necessary for a statistical description hold for these systems, experiments will be necessary for deciding their applicability.

While computational and biological systems such as social insects and multicellular organisms share a number of features, we should also note there are a number of important differences. For instance, in contrast to biological individuals, computational agents are programmed to complete their tasks as soon as possible, which in turn implies a desirability for their earliest death. This task completion may also involve terminating other processes spawned to work on different aspects of the same problem, as

in parallel search, where the first process to find a solution terminates the others. This much more rapid turnover of agents can be expected to lead to dynamics at much shorter time scales than seen in biological or economic counterparts.

Another interesting difference between biological and computational ecologies lies in the fact that for the latter the local rules (or programs for the processes) can be arbitrarily defined, whereas in biology those rules are quite fixed. Moreover, in distributed computational systems the interactions are not constrained by a Euclidean metric, so that processes separated by large physical distances can strongly affect each other by passing messages of arbitrary complexity between them. And last but not least, in computational ecologies the rationality assumption of game theory can be explicitly imposed on their agents, thereby making these systems amenable to game dynamic analysis, suitably adjusted for their intrinsic characteristics. On the other hand, computational agents are considerably less sophisticated in their decision making capacity than people, which could prevent expectations based on observed human performance from being realized.

There are by now a number of distributed computational systems which exhibit many of the above characteristics, and that offer increased performance when compared with traditional operating systems. For instance a number of market based systems have been developed [3]. *Enterprise* [12] is a market-like scheduler where independent processes or agents are allocated at run time among remote idle workstations through a bidding mechanism. A more evolved system, *Spawn* [16], is organized as a market economy composed of interacting buyers and sellers. The commodities in this economy are computer processing resources; specifically, slices of CPU time on various types of computers in a distributed computational environment. The system has been shown to provide substantial improvements over more conventional systems, while providing dynamic response to changes and resource sharing.

Another interesting application of distributed control for autonomous systems is *smart matter*. These are mechanical systems with embedded microscopic sensors, computers and actuators that actively monitor and respond to their environments in precisely controlled ways. These are microelectromechanical systems (MEMS) [2, 1] where the devices are fabricated together in single silicon wafers. A robust control approach for such systems uses a collection of distributed autonomous processes, or *agents*, that each deal with a limited part of the overall control problem [8]. Individual agents can be associated with each sensor or actuator in the material, or with various aggregations of these devices, to provide a mapping between agents and physical location.

From a scientific point of view, the analogy between distributed computation and natural ecologies brings to mind the spontaneous appearance of organized behavior in biological and social systems, where agents can engage in cooperating strategies while working on the solution of particu-

lar problems. In some cases, the strategy mix used by these agents evolves towards an asymptotic ratio that is constant in time and stable against perturbations. This phenomenon sometimes goes under the name of evolutionarily stable strategy (ESS). Recently, it has been shown that spontaneous organization can also exist in open computational systems when agents can choose among many possible strategies while collaborating in the solution of computational tasks. In this case however, imperfect knowledge and delays in information introduce asymptotic oscillatory and chaotic states that exclude the existence of simple ESS's. This is an important finding in light of studies that resort to notions of evolutionarily stable strategies in the design and prediction of open system's performance.

In what follows we will describe a market based computational ecosystem and a theory of distributed computation. The theory describes the collective dynamics of computational agents, while incorporating many of the features endemic to such systems, including distributed control, asynchrony, resource contention, and extensive communication among agents. When processes can choose among many possible strategies while collaborating in the solution of computational tasks, the dynamics leads to asymptotic regimes characterized by complex attractors. Detailed experiments have confirmed many of the theoretical predictions, while uncovering new phenomena, such as chaos induced by overly clever decision-making procedures.

Next, we will deal with the problem of controlling chaos in such systems, for we have discovered ways of achieving global stability through local controls inspired by fitness mechanisms found in nature. Furthermore, we will show how diversity enters into the picture, along with the minimal amount of such diversity that is required to achieve stable behavior in a distributed computational system.

13.2 Computational Markets for Resource Allocation

Allocating resources to competing tasks is one of the key issues for making effective use of computer networks. Examples include deciding whether to run a task in parallel on many machines or serially on one; and whether to save intermediate results or recompute them as needed. The similarity of this problem to resource allocation in market economies, has prompted considerable interest in using analogous techniques to schedule tasks in a network environment. In effect, a coordinated solution to the allocation problem is obtained using Adam Smith's "invisible hand" [14]. Although unlikely to produce the optimal allocation that would be made by an omniscient controller with unlimited computational capability, it can perform well compared to other feasible alternatives [4, 11]. As in economics [6],

the use of prices provides a flexible mechanism for allocating resources, with relatively low information requirements: a single price summarizes the current demand for each resource, whether processor time, memory, communication bandwidth, use of a database or control of a particular sensor. This flexibility is especially desirable when resource preferences and performance measures differ among tasks. For instance an intensive numerical simulation's need for fast floating point hardware is quite different from an interactive text editor's requirement for rapid response to user commands or a database search's requirement for rapid access to the data and fast query matching.

As a conceptual example of how this could work in a computational setting, suppose that a number of database search tasks are using networked computers to find items of interest to various users. Furthermore, suppose that some of the machines have fast floating point hardware but all are otherwise identical. Assuming the search tasks make little use of floating point operations, their performance will not depend on whether they run on a machine with fast floating point hardware. In a market based system, these programs will tend to value each machine based on how many other tasks it is running, leading to a uniform load on the machines. Now suppose some floating point intensive tasks arrive in the system. These will definitely prefer the specialized machines and consequently bid up the price of those particular resources. The database tasks, observing that the price for some machines has gone up, will then tend to migrate toward those machines without the fast floating point hardware. Importantly, because of the high cost of modifying large existing programs, the database tasks will not need to be rewritten to adjust for the presence of the new tasks. Similarly, there is no need to reprogram the scheduling method of a traditional central controller, which is often very time consuming.

This example illustrates how a reasonable allocation of resources could be brought about by simply having the tasks be sensitive to current resource price. Moreover, adjustments can take place continually as new uses are found for particular network resources (which could include specialized databases or proprietary algorithms as well as the more obvious hardware resources), and do not require all users to agree on, or even know about, these new uses, thus encouraging an incremental and experimental approach to resource allocation.

While this example motivates the use of market based resource allocation, a study of actual implementations is required to see how large the system must be for its benefits to appear and whether any of the differences between simple computer programs and human agents pose additional problems. In particular, a successful use of markets requires a number of changes to traditional computer systems. First the system must provide an easily accessible, reliable market so that buyers and sellers can quickly find each other. Second, individual programs must be price sensitive so they will respond to changes in relative prices among resources. This implies that

the programs must, in some sense at least, be able to make choices among various resources based on how well suited they are for the task at hand.

A number of market-like systems have been implemented over the years [12, 15, 16]. Most instances focus on finding an appropriate machine for running a single task. While this is important, further flexibility is provided by systems that use market mechanisms to also manage a collection of parallel processes contributing to the solution of a single task. In this latter case, prices give a flexible method for allocating resources among multiple competing heuristics for the same problem based on their perceived progress. It thus greatly simplifies the development of programs that adjust to unpredictable changes in resource demand or availability. Thus we have a second reason to consider markets: not only may they be useful for flexible allocation of computational resources among competing tasks, but also the simplicity of the price mechanism could provide help with designing cooperative parallel programs.

One such system is Spawn [16], in which each task, starting with a certain amount of money corresponding to its relative priority, bids for the use of machines on the network. In this way, each task can allocate its budget toward those resources most important for it. In addition, when prices are low enough, some tasks can split into several parts which run in parallel, as shown in Fig. 13.1, thereby adjusting the number of machines devoted to each task based on the demand from other users. From a user's point of view, starting a task with the Spawn system amounts to giving a command to execute it and the necessary funding for it to buy resources. The Spawn system manages auctions on each of the participating machines, the use of resources by each participating task, and provides communication paths among the spawned processes. It remains for the programmer to determine the specific algorithms to be used and the meaningful subtasks into which to partition the problem. That is, the Spawn system provides the price information and a market, but the individual programs must be written to make their own price decisions to effectively participate in the market. To allow existing, non-price sensitive, programs to run within the Spawn system without modification, we provided a simple default manager that simply attempted to buy time on a single machine for that task. Users could then gradually modify this manager for their particular task, if desired, to spawn subtasks or use market strategies more appropriate for the particular task.

Studies with this system show that an equilibrium price can be meaningfully defined with even a few machines participating. A specific instance is shown in Fig. 13.2. Despite the continuing fluctuations, this small network reaches a rough price equilibrium. Moreover, the ratio of prices between the two machines closely matches their relative speeds, which was the only important difference between the two types of machine for these tasks. An additional experiment studied a network with some lengthy, low priority tasks to which was added a short, high priority task. The new task rapidly

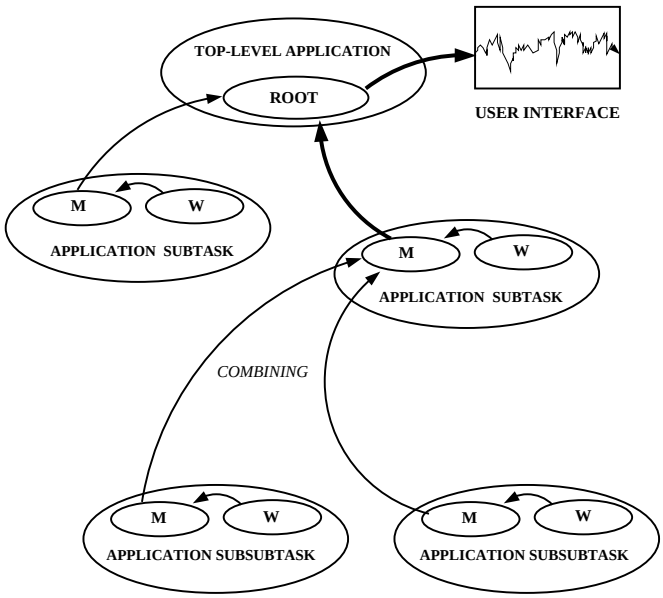

FIGURE 13.1. Managing parallel execution of subtasks in Spawn. Worker processes (W) report progress to their local managers (M) who in turn make reports to the next higher level of management. Upper management combines data into aggregate reports. Finally, the root manager presents results to the user. Managers also bid for the use of additional machines and, if successful, spawn additional subtasks on them.

expands throughout the network by outbidding the existing tasks and driving the price of CPU time up, as shown in Fig. 13.3. It is therefore able to briefly utilize a large number of networked machines and illustrates the inherent flexibility of market based resource allocation. Although the very small networks used in these experiments could be adequately managed centrally, these results do show that expected market behavior can emerge even in small cases.

Computer market systems can be used to experimentally address a number of additional issues. For instance, understanding what happens when more sophisticated programs begin to use the network, e.g., processes that attempt to anticipate future loads so as to maximize their own resource usage. Such behavior can destabilize the overall system. Another area of interest is the emergence of diversity or specialization from a group of initially similar machines. For example, a machine might cache some of the routines or data commonly used by its processes, giving it a comparative advantage in bids for similar tasks in the future. Ultimately this could result in complex organizational structures embedded within a larger market

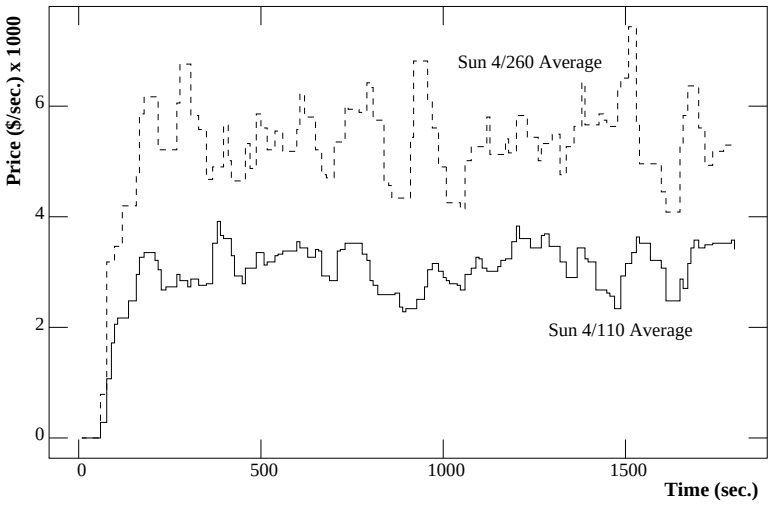

FIGURE 13.2. Price as a function of time (in seconds) in an inhomogeneous Spawn network consisting of three Sun 4/260's and six Sun 4/110's running four independent tasks. The average price of the 260's is the dashed line, the less powerful 110's are solid.

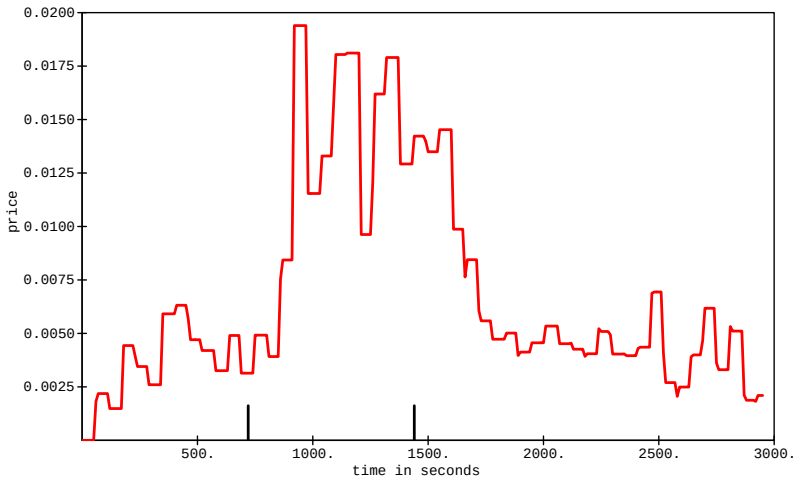

FIGURE 13.3. Price as a function of time (in seconds) when a high-priority task is introduced into a Spawn network running low-priority jobs. The first vertical line segment on the time axis marks the introduction of the high-priority task, and the second one the termination of its funding.

framework [13]. Within these groups, some machines could keep track of the kinds of problems for which others perform best and use this information to guide new tasks to appropriate machines. In this way the system could gradually learn to perform common tasks more effectively.

These experiments also highlighted a number of more immediate practical issues. In setting up Spawn, it was necessary to find individuals willing to allow their machines to be part of the market. While it would seem simple enough to do so, in practice a number of incentives were needed to overcome the natural reluctance of people to have other tasks running on their machines. This reluctance is partly based on perceived limitations on the security of the network and the individual operating systems; for it was possible that a remote procedure could crash an individual machine or consume more resources than anticipated. In particular, users with little need for compute-intensive tasks saw little benefit from participating since they had no use for the money collected by their machines. This indicates the need to use real money in such situations so that these users could use their revenues for their own needs. This in turn, brings the issue of computer security to the forefront so users will feel confident that no counterfeiting of money takes place and tasks will in fact be limited to only use resources they have paid for.

Similarly, for those users participating in the system as buyers, they need to have some idea of what amount of money is appropriate to give a task. In a fully developed market, there could easily be tools to monitor the results of various auctions and hence give a current market price for resources. However, when using a newly created market with only a few users, tools are not always available to give easy access to prices, and even if they are, the prices have large fluctuations. Effective use of such a system also requires users to have some idea of what resources are required for their programs, or, better yet, to encode that information in the program itself so it will be able to respond to available resources, e.g., by spawning subtasks, more rapidly than the users can. Conversely, there must be a mechanism whereby sellers can make available information about the characteristics of their resources (e.g., clock speed, available disk space or special hardware). This can eventually allow for more complex market mechanisms, such as auctions that attempt to sell simultaneous use of different resources (e.g., CPU time and fast memory) or future use of currently unavailable resources to give tasks a more predictable use of resources. Developing and evaluating a variety of auction and price mechanisms that are particularly well suited to these computational tasks is an interesting open problem.

Finally, these experimental systems help clarify the differences between human and computer markets. For instance, computational processes can respond to events much more rapidly than people, but are far less sophisticated. Moreover, unlike the situation with people, particular incentive structures, rationality assumptions, etc. can be explicitly built into computational processes allowing for the possibility of designing particular market

mechanisms. This could lead to the ironic situation in which economic theory has greater predictability for the behavior of computational markets than for that of the larger, and more complex, human economy.

13.3 Chaos in Computational Ecosystems

The Spawn system highlights the need to understand the dynamical behaviors of simple agents with fast response times, compared to human in economic settings, which are complex and slower. To this end we present a dynamical model of resource contention in the presence of imperfect information and delays [9].

In this model, agents independently and asynchronously select among the available choices based on their perceived payoff. These payoffs are actual computational measures of performance, such as the time required to complete a task, accuracy of the solution, amount of memory required, etc. In general, the payoff G_r for using resource r depends on the number of agents already using it. In a purely competitive environment, the payoff for using a particular resource tends to decrease as more agents make use of it. Alternatively, the agents using a resource could assist one another in their computations, as might be the case if the overall task could be decomposed into a number of subtasks. If these subtasks communicate extensively to share partial results, the agents will be better off using the same computer rather than running more rapidly on separate machines and then being limited by slow communications. As another example, agents using a particular database could leave index links that are useful to others. In such cooperative situations, the payoff of a resource would then increase as more agents use it, until it became sufficiently crowded.

Imperfect information about the state of the system causes each agent's perceived payoff to differ from the actual value, with the difference increasing when there is more uncertainty in the information available to the agents. This type of uncertainty concisely captures the effect of many sources of errors such as some program bugs, heuristics incorrectly evaluating choices, errors in communicating the load on various machines and mistakes in interpreting sensory data. Specifically, the perceived payoffs are taken to be normally distributed, with standard deviation σ, around their correct values. In addition, information delays cause each agent's knowledge of the state of the system to be somewhat out of date. Although for simplicity we will consider the case in which all agents have the same effective delay, uncertainty, and preferences for resource use, we should mention that the same range of behaviors is also found in more general situations [7].

As a specific illustration of this approach, we consider the case of two resources so the system can be described by $P(n,t)$, the probability to have n agents selecting the first resource at time t. Its dynamics over a small

time interval Δt is governed by [9]

$$\frac{P(n,t+\Delta t)-P(n,t)}{\Delta t}=\sum_{n'}\left(W(n|n')P(n',t)-W(n'|n)P(n,t)\right) \tag{13.3.1}$$

where $W(n'|n)$ is the transition probability per unit time that the state changes from n to n'. To derive an expression for the transition rates, notice that in the time interval Δt, the probabilities a given agent using resource 1 switches to resource 2 and vice versa are given by

$$\begin{aligned} P(2\rightarrow 1) &= \alpha\Delta t\,\rho \\ P(1\rightarrow 2) &= \alpha\Delta t\,(1-\rho) \end{aligned} \tag{13.3.2}$$

where α is the rate at which agents reevaluate their resource choice and ρ is the probability an agent prefers resource 1 over 2. For a system with network externalities, ρ depends on the number of agents using each resource.

For very short time intervals, the asynchronous decisions mean one can assume that only one of the agents reevaluates the option to switch. This means that $W(n|n')=0$ unless $n-n'=0$, 1 or -1. Taking these three cases into account, the transition probabilities become

$$\begin{aligned} W(n|n')\Delta t &= \delta_{n,n'-1}(n'\alpha\Delta t(1-\rho(n'))) \\ &\quad +\delta_{n,n'+1}((N-n')\alpha\Delta t\rho(n')) \\ &\quad +\delta_{n,n'}(1-n'\alpha\Delta t(1-\rho(n'))-(N-n')\alpha\Delta t\rho(n')) \end{aligned} \tag{13.3.3}$$

As $\Delta t\rightarrow 0$, the probability of any changes in the agent choices goes to zero, making $W(n|n)\Delta t\rightarrow 1$. In this limit, the right-hand side of the equation correctly gives 1 for $n=n'$. Substituting this into Eq. (13.3.1) and letting $\Delta t\rightarrow 0$ gives

$$\begin{aligned} \frac{\partial P(n,t)}{\partial t} &= \alpha P(n,t)[-n(1-\rho(n))-(N-n)\rho(n)] \\ &\quad +\alpha P(n+1,t)[(n+1)(1-\rho(n+1))] \\ &\quad +\alpha P(n-1,t)[(N-n+1)\rho(n-1)] \end{aligned} \tag{13.3.4}$$

We can now compute the dynamics for the average number of agents by using the identity

$$\frac{d}{dt}\langle n\rangle=\sum_{n'=0}^{N}n'\frac{\partial P(n',t)}{\partial t} \tag{13.3.5}$$

Using Eq. (13.3.1) to evaluate the sum on the right hand side gives the evolution of the average number of agents as

$$\begin{aligned} \frac{d\langle n\rangle}{dt} &= \alpha\left\langle (N-n)\rho(n)-n(1-\rho(n))\right\rangle \\ &= \alpha(N\langle\rho(n)\rangle-\langle n\rangle) \end{aligned} \tag{13.3.6}$$

The first line is simply interpreted as the difference between the number of agents using resource 2 who switch to resource 1 and those using resource 1 who switch to resource 2. This expression can be further simplified by invoking the mean field approximation $\langle \rho(n) \rangle \approx \rho(\langle n \rangle)$, and defining the fraction $f \equiv n/N$ of agents which are using resource 1 at any given time:

$$\frac{df}{dt} = \alpha(\rho - f) \tag{13.3.7}$$

With this formulation, it is convenient to treat ρ as a function of f. It can be expressed in terms of the payoffs G_1, G_2 associated with each resource and the uncertainty in the information available to the agents. Specifically for a normally distributed error around the true value of the payoff, ρ becomes

$$\rho = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{G_1(f) - G_2(f)}{2\sigma}\right)\right) \tag{13.3.8}$$

where σ quantifies the uncertainty. Notice that this definition captures the simple requirement that an agent is more likely to prefer a resource when its payoff is relatively large. Finally, delays in information are modelled by supposing that the payoffs that enter into ρ at time t are the values they had at a delayed time $t - \tau$.

For a typical system of many agents with a mixture of cooperative and competitive payoffs, the kinds of dynamical behaviors exhibited by the model are shown in Fig. 13.4. When the delays and uncertainty are fairly small, the system converges to an equilibrium point close to the optimal obtainable by an omniscient, central controller. As the information available to the agents becomes more corrupted, the equilibrium point moves further from the optimal value. With increasing delays, the equilibrium eventually becomes unstable, leading to the oscillatory and chaotic behavior shown in the figure. In these cases, the number of agents using particular resources continues to vary so that the system spends relatively little time near the optimal value, with a consequent drop in its overall performance. This can be due to the fact that chaotic systems are unpredictable, hence making it difficult for individual agents to automatically select the best resources at any given time.

13.4 The Uses of Fitness

We will now describe an effective procedure for controlling chaos in distributed systems [7]. It is based on a mechanism that rewards agents according to their actual performance. As we shall see, such an algorithm leads to the emergence of a diverse community of agents out of an essentially homogenous one. This diversity in turn eliminates chaotic behavior

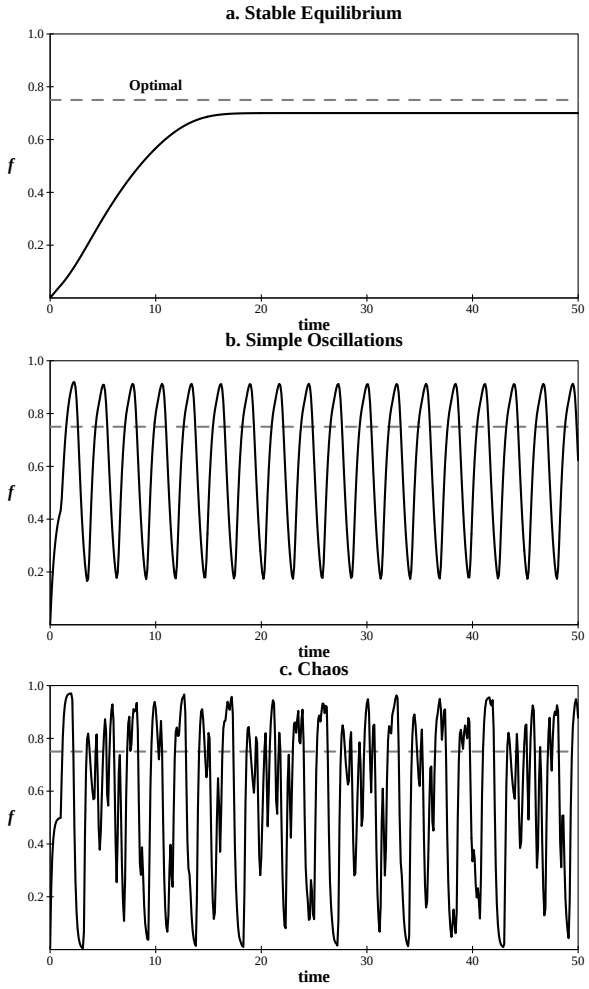

FIGURE 13.4. Typical behaviors for the fraction f of agents using resource 1 as a function of time for successively longer delays: a) relaxation toward stable equilibrium, b) simple persistent oscillations, and c) chaotic oscillations. The payoffs are $G_1 = 4 + 7f - 5.333f^2$ for resource 1 and $G_2 = 4 + 3f$ for resource 2. The time scale is in units of the delay time τ, $\sigma = 1/4$ and the dashed line shows the optimal allocation for these payoffs.

through a series of dynamical bifurcations which render chaos a transient phenomenon.

The actual performance of computational processes can be rewarded in a number of ways. A particularly appealing one is to mimic the mechanism found in biological evolution, where fitness determines the number of survivors of a given species in a changing environment. This mechanism

is used in computation under the name of *genetic algorithms* [5]. Another example is provided by computational systems modelled on ideal economic markets [13, 16], which reward good performance in terms of profits. In this case, agents pay for the use of resources, and they in turn are paid for completing their tasks. Those making the best choices collect the most currency and are able to outbid others for the use of resources. Consequently they come to dominate the system.

While there is a range of possible reward mechanisms, their net effect is to increase the proportion of agents that are performing successfully, thereby decreasing the number of those who do not do as well. It is with this insight in mind that we developed a general theory of effective reward mechanisms without resorting to the details of their implementations. Since this change in agent mix will in turn change the choices made by every agent and their payoffs, those that were initially most successful need not be so in the future. This leads to an evolving diversity whose eventual stability is by no means obvious.

Before proceeding with the theory we point out that the resource payoffs that we will consider are instantaneous ones (i.e., shorter than the delays in the system), e.g., work actually done by a machine, currency actually received, etc. Other reward mechanisms, such as those based on averaged past performance, could lead to very different behavior from the one exhibited in this paper.

In order to investigate the effects of rewarding actual performance we generalize the previous model of computational ecosystems by allowing agents to be of different types, a fact which gives them different performance characteristics. Recall that the agents need to estimate the current state of the system based on imperfect and delayed information in order to make good choices. This can be done in a number of ways, ranging from extremely simple extrapolations from previous data to complex forecasting techniques. The different types of agents then correspond to the various ways in which they can make these extrapolations.

Within this context, a computational ecosystem can be described by specifying the fraction of agents, f_{rs} of a given type s using a given resource r at a particular time. We will also define the total fraction of agents using a resource of a particular type as

$$\begin{aligned} f_r^{\text{res}} &= \sum_s f_{rs} \\ f_s^{\text{type}} &= \sum_r f_{rs} \end{aligned} \tag{13.4.9}$$

respectively.

As mentioned previously, the net effect of rewarding performance is to increase the fraction of highly performing agents. If γ is the rate at which performance is rewarded, then Eq. (13.3.7) is enhanced with an extra term

which corresponds to this reward mechanism. This gives

$$\frac{df_{rs}}{dt} = \alpha \left(f_s^{\text{type}} \rho_{rs} - f_{rs} \right) + \gamma \left(f_r^{\text{res}} \eta_s - f_{rs} \right) \tag{13.4.10}$$

where the first term is analogous to that of the previous theory, and the second term incorporates the effect of rewards on the population. In this equation ρ_{rs} is the probability that an agent of type s will prefer resource r when it makes a choice, and η_s is the probability that new agents will be of type s, which we take to be proportional to the actual payoff associated with agents of type s. As before, α denotes the rate at which agents make resource choices and the detailed interpretation of γ depends on the particular reward mechanism involved. For example, if they are replaced on the basis of their fitness it is the rate at which this happens. In a market system, on the other hand, γ corresponds to the rate at which agents are paid. Notice that in this case, the fraction of each type is proportional to the wealth of agents of that type.

Since the total fraction of agents of all types must be one, a simple form of the normalization condition can be obtained if one considers the relative payoff, which is given by

$$\eta_s = \frac{\sum_r f_{rs} G_r}{\sum_r f_r^{\text{res}} G_r} \tag{13.4.11}$$

Note that the numerator is the actual payoff received by agents of type s given their current resource usage and the denominator is the total payoff for all agents in the system, both normalized to the total number of agents in the system. This form assumes positive payoffs, e.g., they could be growth rates. If the payoffs can be negative (e.g., they are currency changes in an economic system), one can use instead the difference between the actual payoffs and their minimum value m. Since the η_s must sum to 1, this will give

$$\eta_s = \frac{\sum_r f_{rs} G_r - m}{\sum_r f_r^{\text{res}} G_r - Sm} \tag{13.4.12}$$

which reduces to the previous case when $m = 0$.

Summing Eq. (13.4.10) over all resources and types gives

$$\begin{aligned} \frac{df_r^{\text{res}}}{dt} &= \alpha \left(\sum_s f_s^{\text{type}} \rho_{rs} - f_r^{\text{res}} \right) \\ \frac{df_s^{\text{type}}}{dt} &= \gamma \left(\eta_s - f_s^{\text{type}} \right) \end{aligned} \tag{13.4.13}$$

which describe the dynamics of overall resource use and the distribution of agent types, respectively. Note that this implies that those agent types which receive greater than average payoff (i.e., types for which $\eta_s > f_s^{\text{type}}$) will increase in the system at the expense of the low performing types.

Note that the actual payoffs can only reward existing types of agents. Thus in order to introduce new variations into the population an additional mechanism is needed (e.g., corresponding to mutation in genetic algorithms or learning).

13.5 Results

In order to illustrate the effectiveness of rewarding actual payoffs in controlling chaos, we examine the dynamics generated by Eq. (13.4.10) for the case in which agents choose among two resources with cooperative payoffs, a case which we have shown to generate chaotic behavior in the absence of rewards [9, 10]. As in the particular example of Fig. 13.4c, we use $\tau = 10$, $G_1 = 4 + 7f_1 - 5.333f_1^2$, $G_2 = 7 - 3f_2$, $\sigma = 1/4$ and an initial condition in which all agents start by using resource 2.

One kind of diversity among agents is motivated by the simple case in which the system oscillates with a fixed period. In this case, those agents that are able to discover the period of the oscillation can then use this knowledge to reliably estimate the current system state in spite of delays in information. Notice that this estimate does not necessarily guarantee that they will keep performing well in the future, for their choice can change the basic frequency of oscillation of the system.

In what follows, we take the diversity of agent types to correspond to the different past horizons, or extra delays, that they use to extrapolate to the current state of the system. These differences in estimation could be due to having a variety of procedures for analyzing the system's behavior. Specifically, we identify different agent types with the different assumed periods which range over a given interval. Thus, we take agents of type s to use an effective delay of $\tau + s$ while evaluating their choices.

The resulting behavior is shown in Fig. 13.5 which should be contrasted with Fig. 13.4c. We used an interval of extra delays ranging from 0 to 40. As shown, the introduction of actual payoffs induces a chaotic transient which, after a series of dynamical bifurcations, settles into a fixed point that signals stable behavior. Furthermore, this fixed point is exactly that obtained in the case of no delays. That this equilibrium is stable against perturbations can seen by the fact that if the system were perturbed again (as shown in Fig. 13.6), it rapidly returns to its previous value. In additional experiments, with a smaller range of delays, we found that the system continued to oscillate without achieving the fixed point.

This transient chaos and its eventual stability can be understood from the distribution of agents with extra delays as a function of time. As can be seen in Fig. 13.7 actual payoffs lead to a highly heterogeneous system, characterized by a diverse population of agents of different types. It also shows that the fraction of agents with certain extra delays increases greatly.

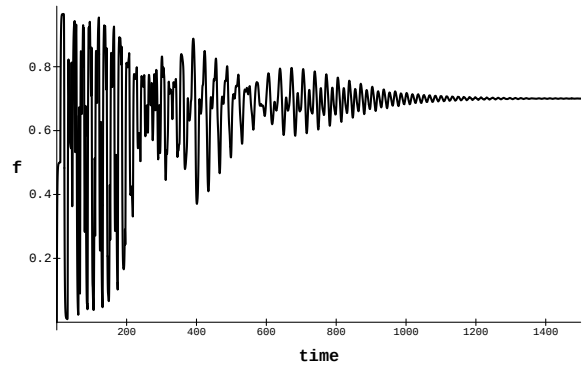

FIGURE 13.5. Fraction of agents using resource 1 as a function of time with adjustment based on actual payoff. These parameters correspond to Fig. 13.4c so without the adjustment the system would remain chaotic.

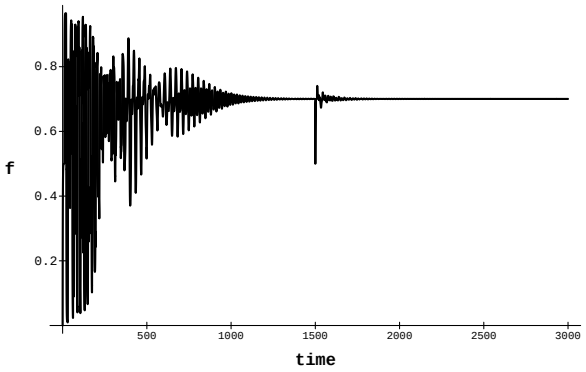

FIGURE 13.6. Behavior of the system shown in Fig. 13.5 with a perturbation introduced at time 1500.

These delays correspond to the major periodicities in the system.

13.6 Stability and Minimal Diversity

As we showed in the previous section, rewarding the performance of large collections of agents engaging in resource choices leads to a highly diverse mix of agents that stabilize the system. This suggests that the real cause of stability in a distributed system is that provided by sufficient diversity, and that the reward mechanism is an efficient way of automatically finding a good mix. This raises the interesting question of the minimal amount of diversity needed in order to have a stable system.

The stability of a system is determined by the behavior of a perturba-

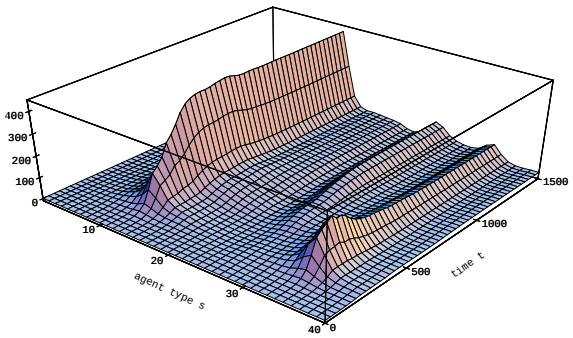

FIGURE 13.7. Ratio $f_s^{\rm type}(t)/f_s^{\rm type}(0)$ of the fraction of agents of each type, normalized to their initial values, as a function of time. Note there are several peaks, which correspond to agents with extra delays of 12, 26 and 34 time units. Since $\tau = 10$, these match periods of length 22, 36 and 44 respectively.

tion around equilibrium, which can be found from the linearized version of Eq. (13.4.10). In our case, the diversity is related to the range of different delays that agents can have. For a continuous distribution of extra delays, the characteristic equation is obtained by assuming a solution of the type $e^{\lambda t}$ in the linearized equation, giving

$$\lambda + \alpha - \alpha\rho' \int ds\, f(s) e^{-\lambda(s+\tau)} = 0 \tag{13.6.14}$$

Stability requires that all the values of λ have negative real parts, so that perturbations will relax back to equilibrium. As an example, suppose agent types are uniformly distributed in $(0, S)$. Then $f(s) = 1/S$, and the characteristic equation becomes

$$\lambda + \alpha - \alpha\rho' \frac{1 - e^{-\lambda S}}{\lambda S} e^{-\lambda\tau} = 0 \tag{13.6.15}$$

Defining a normalized measure of the diversity of the system for this case by $\eta \equiv S/\tau$, introducing the new variable $z \equiv \lambda\tau(1+\eta)$, and multiplying Eq. (13.6.15) by $\tau(1+\eta)ze^z$ introduces an extra root at $z = 0$ and gives

$$(z^2 + az)e^z - b + be^{rz} = 0 \tag{13.6.16}$$

where

$$\begin{aligned} a &= \alpha\tau(1+\eta) > 0 \\ b &= -\rho' \frac{\alpha\tau(1+\eta)^2}{\eta} > 0 \\ r &= \frac{\eta}{1+\eta} \in (0,1) \end{aligned} \tag{13.6.17}$$

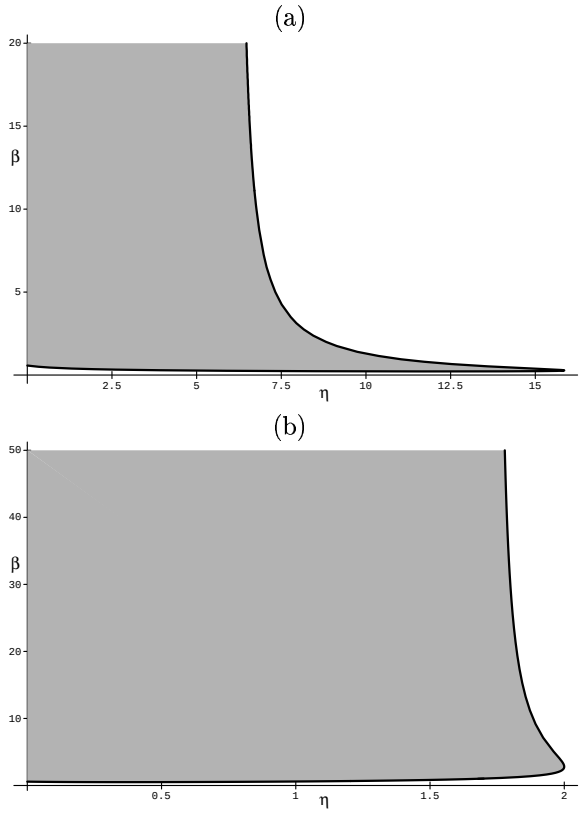

FIGURE 13.8. Stability as a function of $\beta = \alpha\tau$ and $\eta = S/\tau$ for two possible distributions of agent types: a) $f(s) = 1/S$ in $(0, S)$, and b) $f(s) = (1/S)e^{-s/S}$. The system is unstable in the shaded regions and stable to the right and below the curves.

The stability of the system with uniform distribution of agents with extra delays thus reduces to finding the condition under which all roots of Eq. (13.6.16), other than $z = 0$, have negative real parts. This equation is a particular instance of an *exponential polynomial*, whose terms consist of powers multiplied by exponentials. Unlike regular polynomials, these objects generally have an infinite number of roots, and are important in the study of the stability properties of differential-delay equations. Established methods can then be used to determine when they have roots with positive real parts. This in turn defines the stability boundary of the equation. The result for the particular case in which $\rho' = -3.41044$, corresponding to the parameters used in Section 13.5, is shown in the left half of Fig. 13.8.

Similarly, if we choose an exponential distribution of delays, i.e., $f(s) =$

$(1/S)e^{-s/S}$ with positive S, the characteristic equation acquires the form

$$(z^2 + pz + q)e^z + r = 0 \tag{13.6.18}$$

where

$$\begin{aligned} p &= \alpha\tau + \frac{1}{\eta} > 0 \\ q &= \frac{\alpha\tau}{\eta} > 0 \\ r &= -\frac{\alpha\tau\rho'}{\eta} > 0 \end{aligned} \tag{13.6.19}$$

and $z \equiv \lambda\tau$. An analysis similar to that for the uniform distribution case leads to the stability diagram shown in the right hand side of the figure.

Although the actual distributions of agent types can differ from these two cases, the similarity between the stability diagrams suggests that regardless of the magnitude of β one can always find an appropriate mix that will make the system stable. This property follows from the vertical asymptote of the stability boundary. It also illustrates the need for a minimum diversity in the system in order to make it stable when the delays aren't too small.

Having established the right mix that produces stability one may wonder whether a static assignment of agent types at an initial time would not constitute a simpler and more direct procedure to stabilize the system without resorting to a dynamic reward mechanism. While this is indeed the case in a non-fluctuating environment, such a static mechanism cannot cope with changes in both the nature of the system (e.g., machines crashing) and the arrival of new tasks or fluctuating loads. It is precisely to avoid this vulnerability by keeping the system adaptive that a dynamic procedure is needed.

Having seen how sufficient diversity stabilizes a distributed system, we now turn to the mechanisms that can generate such heterogeneity, as well as the time that it takes for the system to stabilize. In particular, the details of the reward procedures determine whether the system can even find a stable mix of agents. In the cases describe above, reward was proportional to actual performance, as measured by the payoffs associated with the resources used. One might also wonder whether stability would be achieved more rapidly by giving greater (than their fair share) increases to the top performers.

We have examined two such cases: a) rewards proportional to the square of their actual performance, and b) giving all the rewards to top performers (e.g., those performing at the 90th percentile or better in the population). In the former case we observed stability with a shorter transient, whereas in the latter case the mix of changes continued to change through time, thus preventing stable behavior. This can be understood in terms of our earlier observation that whereas a small percentage agents can identify oscillation

periods and thereby reduce their amplitude, a large number of them can no longer perform well.

Note that the time to reach equilibrium is determined by two parameters of the system. The first is the time that it takes to find a stable mix of agent types, which is governed by γ, and the second the rate at which perturbations relax, given the stable mix. The latter is determined by the largest real part of any of the roots, λ, of the characteristic equation.

13.7 Discussion

In this paper we have presented a case for treating distributed computation as an ecosystem, an analogy that turns out to be quite fruitful in the analysis, design, and control of such systems. In spite of the many differences between computational processes and organisms, resource contention, complex dynamics and reward mechanisms seem to be ubiquitous in distributed computation, making it also a tool for the study of natural ecosystems.

Since chaotic behavior seems to be the natural resultant of interacting processes with imperfect and delayed information, the problem of controlling such systems is of paramount importance. We discovered that rewards based on the actual performance of agents in a distributed computational system can stabilize an otherwise chaotic or oscillatory system. This leads in turn to greatly improved system performance.

In all these cases, stability is achieved by making chaos a transient phenomena. In the case of distributed systems, the addition of the reward mechanism has the effect of dynamically changing the control parameters of the resource allocation dynamics in such a way that a global fixed point of the system is achieved. This brings the issue of the length of the chaotic transient as compared to the time needed for most agents to complete their tasks. Even when the transients are long, the results of this study show that the range gradually decreases, thereby improving performance even before the fixed point is achieved.

A particularly relevant question for distributed systems is the extent to which these results generalize beyond the mechanism that we studied. We considered the specific situation of a collection of agents with different delays in their appraisal of the system evolution. Similar behavior is also observed if the agents have a bias for a particular resource. It is of interest to inquire whether using rewards to increase diversity works more generally than in these cases.

Since we only considered agents choosing between only two resources, it is important to understand what happens when there are many resources the agents can choose from. One may argue that since diversity is the key to stability, a plurality of resources provides enough channels to develop the necessary heterogeneity, which is what we observed in situations with three

resources. Another note of caution has to do with the effect of fluctuations on a finite population of agent types. While we have shown that sufficient diversity can, on average, stabilize the system, in practice a fluctuation could wipe out those agent types that would otherwise be successful in stabilizing the system. Thus, we need either a large number of each kind of agent or a mechanism, such as mutation, to create new kinds of agents.

Another issue concerns the time scales over which rewards are assigned to agents. In our treatment, we assumed the rewards were always based on the performance at the time they were given. Since in many cases this procedure is delayed, there is the question of the extent to which rewards based on past performance are also able to stabilize chaotic distributed systems.

Finally the validity of this approach will have to be determined by actual implementations and measurements of distributed systems. This will present some challenges in identifying the relevant variables to be measured and aggregated to correspond to quantities used in the theory.

The fact that these simple resource allocation mechanisms work and produce a stable environment provides a basis for developing more complex software systems that can be used for a wide range of computational problems.

13.8 References

[1] A. A. Berlin, H. Abelson, N. Cohen, L. Fogel, C. M. Ho, M. Horowitz, J. How, T. F. Knight, R. Newton, and K. Pister. Distributed information systems for MEMS. Technical report, Information Systems and Technology (ISAT) Study, 1995.

[2] Janusz Bryzek, Kurt Petersen, and Wendell McCulley. Micromachines on the march. *IEEE Spectrum*, pages 20–31, May 1994.

[3] Scott H. Clearwater, editor. *Market-Based Control: A Paradigm for Distributed Resource Allocation.* World Scientific, Singapore, 1996.

[4] Donald Ferguson, Yechiam Yemini, and Christos Nikolaou. Microeconomic algorithms for load balancing in distributed computer systems. In *International Conference on Distributed Computer Systems*, pages 491–499. IEEE, 1988.

[5] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, NY, 1989.

[6] Friedrich A. Hayek. Competition as a discovery procedure. In *New Studies in Philosophy, Politics, Economics and the History of Ideas*, pages 179–190. University of Chicago Press, Chicago, 1978.

[7] Tad Hogg and Bernardo A. Huberman. Controlling chaos in distributed systems. *IEEE Trans. on Systems, Man and Cybernetics*, 21(6):1325–1332, November/December 1991.

[8] Tad Hogg and Bernardo A. Huberman. Controlling smart matter. *Smart Materials and Structures*, 7:R1–R14, 1998. Los Alamos preprint cond-mat/9611024.

[9] Bernardo A. Huberman and Tad Hogg. The behavior of computational ecologies. In B. A. Huberman, editor, *The Ecology of Computation*, pages 77–115. North-Holland, Amsterdam, 1988.

[10] J. O. Kephart, T. Hogg, and B. A. Huberman. Dynamics of computational ecosystems. *Physical Review A*, 40:404–421, 1989.

[11] James F. Kurose and Rahul Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.

[12] T.W. Malone, R. E. Fikes, K. R. Grant, and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. In B. A. Huberman, editor, *The Ecology of Computation*, pages 177–205. North-Holland, Amsterdam, 1988.

[13] Mark S. Miller and K. Eric Drexler. Markets and computation: Agoric open systems. In B. A. Huberman, editor, *The Ecology of Computation*, pages 133–176. North-Holland, Amsterdam, 1988.

[14] Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. Univ. of Chicago Press, Chicago, 1976. Reprint of the 1776 edition.

[15] I. E. Sutherland. A futures market in computer time. *Communications of the ACM*, 11(6):449–451, June 1968.

[16] Carl A. Waldspurger, Tad Hogg, Bernardo A. Huberman, Jeffery O. Kephart, and W. Scott Stornetta. Spawn: A distributed computational economy. *IEEE Trans. on Software Engineering*, 18(2):103–117, February 1992.